



---

**TESIS – CI 2541**

**PENILAIAN KUALITAS APLIKASI WEB DENGAN ISO 9126**

Anita Hidayati

NRP. 5107 201 012

DOSEN PEMBIMBING

Daniel Oranova Siahaan, S.Kom, M.Sc, P.D.Eng  
Sarwosri, S.Kom, M.T

PROGRAM MAGISTER

BIDANG KEAHLIAN TEKNIK INFORMATIKA

JURUSAN TEKNIK INFORMATIKA

FAKULTAS TEKNOLOGI INFORMASI

INSTITUT TEKNOLOGI SEPULUH NOPEMBER

SURABAYA

2009



---

**TESIS – CI 2541**

**ASSESSING THE QUALITY OF WEB APPLICATION BY USING ISO  
9126**

Anita Hidayati

5107 201 012

**SUPERVISOR**

Daniel Oranova Siahaan, S.Kom, M.Sc, P.D.Eng  
Sarwosri, S.Kom, M.T

**PROGRAM MAGISTER**

**BIDANG KEAHLIAN TEKNIK INFORMATIKA**

**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS TEKNOLOGI INFORMASI**

**INSTITUT TEKNOLOGI SEPULUH NOPEMBER**

**SURABAYA**

**2009**

## **PENILAIAN KUALITAS APLIKASI WEB DENGAN ISO 9126**

**Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar  
Magister Komputer (M.Kom.)  
di  
Institut Teknologi Sepuluh Nopember**

Oleh :

**ANITA HIDAYATI**

**NRP. 5107 201 012**

**Tanggal Ujian : 30 Juli 2009  
Periode Wisuda : Oktober 2009**

**Disetujui oleh:**

- |  |                               |
|--|-------------------------------|
| <b>1. Daniel O. Siahaan, S.Kom, M.Sc, P.D.Eng<br/>NIP. 132 318 029</b> | <b>.....<br/>(Pembimbing)</b> |
| <b>2. Sarwosri, S.Kom, M.T<br/>NIP. 132 172 210</b>                    | <b>.....<br/>(Pembimbing)</b> |
| <b>3. Prof.Dr.Ir. Handayani Tjandrasa, M.Sc<br/>NIP. 130 532 048</b>   | <b>.....<br/>(Penguji)</b>    |
| <b>4. Fajar Baskoro, S.Kom, M.T<br/>NIP. 132 230 429</b>               | <b>.....<br/>(Penguji)</b>    |
| <b>5. Chastine Fatichah, S.Kom, M.Kom<br/>NIP. 132 298 829</b>         | <b>.....<br/>(Penguji)</b>    |

**Direktur Program Pascasarjana,**

**Prof. Dr. Ir. Suparno, MSIE  
NIP. 130 532 035**

## **PENILAIAN KUALITAS APLIKASI WEB DENGAN ISO 9126**

Nama mahasiswa : Anita Hidayati  
NRP : 5107 201 012  
Pembimbing : Daniel O. Siahaan, S.Kom, M.Sc, P.D.Eng  
Sarwosri, S.Kom, M.T

### **ABSTRAK**

Penelitian ini bertujuan untuk membuat suatu model penilaian kualitas aplikasi web berdasarkan model kualitas ISO 9126. Akan dibuat CASE Tool yang dapat menilai kualitas aplikasi web dan memberikan rekomendasi untuk perbaikan kualitas. Sehingga memungkinkan pihak pengembang aplikasi web memperoleh penilaian kualitas secara lebih komprehensif.

Penghitungan bobot menggunakan Fuzzy Analytical Hierarchical Process (FAHP). Dari masukan pengguna terhadap tingkat preferensi faktor dan sub faktor kualitas didapatkan matrik perbandingan. Dilakukan konversi ke bilangan fuzzy terlebih dulu sebelum dihitung rata-rata fuzzy (r) dan bobot fuzzy (w) sehingga diperoleh bobot faktor kualitas dan bobot relatif sub faktor kualitas. Bobot absolut dari sub faktor kualitas diperoleh dengan mengalikan bobot faktor kualitas dengan bobot relatif sub faktor kualitas. Nilai total kualitas dari suatu aplikasi adalah hasil perkalian bobot absolut dengan nilai kriteria masukan pengguna. Agar penilaian kualitas aplikasi web menjadi lebih mudah dan terukur, dibuatkan pengguna bagi pengguna dalam mengisi nilai dari kriteria kualitas.

CASE Tool dirancang untuk dapat memberikan saran bagi perbaikan kualitas. Dibuat dua macam rekomendasi, yang pertama rekomendasi umum terhadap keseluruhan sistem. Dicari lima nilai kualitas terendah dari semua sub faktor kualitas setelah dilakukan pengurutan. Kemudian dicari nilai kriteria kualitas dari kelima sub faktor tersebut yang memiliki nilai  $<3$ . Ditampilkan saran perbaikan dari tiap kriteria kualitas. Dicari nilai kriteria kualitas yang memiliki nilai  $<3$ . Kemudian ditampilkan saran perbaikannya setelah diurutkan dari nilai kualitas terendah yang menunjukkan tingkat urgensi yang tinggi.

Dari hasil uji coba, CASE Tool mampu memberikan panduan bagi pengguna dalam melakukan penilaian terhadap kriteria kualitas. Selain telah dapat menilai kualitas aplikasi web, CASE Tool juga memberikan rekomendasi untuk perbaikan kualitas. Pembobotan faktor dan sub faktor kualitas berhasil dilakukan dengan menggunakan FAHP. Kontribusi lainnya adalah adanya pemetaan kriteria kualitas aplikasi web ke sub faktor kualitas dari ISO 9126 sehingga CASE Tool dapat menilai kualitas dengan lebih detail sampai ke level sub faktor kualitas

**Kata kunci: Model Kualitas, ISO 9126, Aplikasi Web, FAHP**

# **ASSESSING THE QUALITY OF WEB APPLICATION BY USING ISO 9126**

By : Anita Hidayati  
Student Identity Number : 5107 201 012  
Supervisor : Daniel O. Siahaan, S.Kom, M.Sc, P.D.Eng  
Sarwosri, S.Kom, M.T

## **ABSTRACTS**

This research is aimed to build a model that assesses the quality of web application based on ISO 9126 quality model. Thus, the model will be applied into CASE tool that can assess the quality of web application and give recommendation to increase its quality. So it will enable the developer to get more comprehensive assessing quality.

The computation of weight uses Fuzzy Analytical Hierarchycal Process (FAHP). Moreover the comparison matrix is found from the user input on the factor preference level and sub quality factor. Before the fuzzy rate ( $r$ ) and fuzzy weight ( $w$ ) are calculated, it was converted to fuzzy number. The fuzzy weight is defuzzified so that it will get the weight of quality factor and relative weight of sub quality factor. The absolute weight of sub quality factor is resulted by multiplying the weight of quality factor and the relative weight of sub quality factor. The application total quality is the result of multiplication between absolute weight and criteria value from the user input. Furthermore, to make it more easier and measurable, CASE TOOL makes the guidance for the user in filling the value of quality criteria.

CASE TOOL is designed to recommend the quality improvement. There are two recommendations; the first is general recommendation of the whole system. It searches the five highest values of the sorted sub quality factors' relative weight. Then, the value of quality criteria that have  $<3$  value will be searched. The improvement recommendation will be shown on every quality criteria. The second is special recommendation on every quality factor of the value of quality criteria that have  $<3$  value. The recommendation will be shown from the sorted lowest value that shows the highest urgency.

The result of testing shows that CASE Tool is able to guide user in assessing the quality criteria. Not only evaluating the quality of web application, CASE Tool is also giving recommendation to improve the quality. By using FAHP, factor and sub quality factor weighing can be successfully carried out. Mapping quality criteria of web application into sub quality factor from ISO 9126 is CASE Tool's other contribution so that it can assess more detailed quality until its sub quality factor level.

**Keywords:** Quality Model, ISO 9126, Web Application, FAHP

## DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGESAHAN	ii
ABSTRAK	iii
ABSTRACT	iv
KATA PENGANTAR	v
DAFTAR ISI	vii
DAFTAR GAMBAR	viii
DAFTAR TABEL	x
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	2
1.3 Tujuan dan Manfaat Penelitian	2
BAB 2 KAJIAN PUSTAKA DAN DASAR TEORI	3
2.1 Kualitas Perangkat Lunak	3
2.2 Model Kualitas Perangkat Lunak	6
2.3 ISO 9126	13
2.4 Aplikasi Web	18
2.5 Fuzzy Analytical Hierarchy Process (FAHP)	23
BAB 3 METODA PENELITIAN	27
3.1 Pembuatan Panduan Penilaian Kualitas	27
3.2 Pembuatan Saran untuk Perbaikan Kualitas	32
3.3 Pemetaan Kriteria ke Sub-Faktor Kualitas	34
3.4 Perancangan CASE tool	36
3.4.1 Perancangan Data	36
3.4.2 Perancangan Alur Proses	38
3.4.3 Perancangan Antar Muka	40
3.4.4 Pembobotan dengan Fuzzy AHP	43
3.4.5 Pembuatan Rekomendasi	45
3.5 Implementasi CASE Tool	46
3.5.1 Implementasi Data	46
3.5.2 Implementasi Antar Muka	48
BAB 4 HASIL PENELITIAN DAN PEMBAHASAN	55
4.1 Uji Coba	55
4.2 Analisa Uji Coba	76
BAB 5 KESIMPULAN DAN SARAN	79
5.1 Kesimpulan	79
5.2 Saran	79
DAFTAR PUSTAKA	81

## **DAFTAR GAMBAR**

Gambar 2.1	Hubungan Kebutuhan dan Karakteristik PL	5
Gambar 2.2	Hubungan Komponen Model Kualitas	8
Gambar 2.3	Model Kualitas ISO 9126	15
Gambar 3.1	Blok Diagram Pengerjaan Penelitian	27
Gambar 3.2	Conceptual Data Model	36
Gambar 3.3	Physical Data Model	37
Gambar 3.4	DFD Level 0 Sistem	38
Gambar 3.5	DFD Level 1 Detail Sistem	39
Gambar 3.6	DFD Level 2 Pembobotan	40
Gambar 3.7	Form Login	40
Gambar 3.8	Form Identitas Aplikasi	41
Gambar 3.9	Form Bobot Faktor Kualitas	41
Gambar 3.10	Form Bobot Sub-Faktor Kualitas	42
Gambar 3.11	Form Kriteria	42
Gambar 3.12	Blok Diagram Pembobotan	43
Gambar 3.13	Form Jadi Login	48
Gambar 3.14	Form Registrasi Login	48
Gambar 3.15	Menu dari Aplikasi	48
Gambar 3.16	Form Jadi Identitas Aplikasi	49
Gambar 3.17	Form Pilih Aplikasi	49
Gambar 3.18	Form Jadi Pembobotan Faktor Kualitas	50
Gambar 3.19	Form Jadi Pembobotan Sub Faktor Kualitas	50
Gambar 3.20	Form Jadi Penilaian Kriteria	51
Gambar 3.21	Sub Menu Help Faktor Kualitas	51
Gambar 3.22	Sub Menu Help Sub Faktor Kualitas	52
Gambar 3.23	Sub Menu Help Kriteria Kualitas	52
Gambar 3.24	Sub Menu Help Korelasi	53
Gambar 4.1	Isian Form Login	55
Gambar 4.2	Isian Form Registrasi Login	56
Gambar 4.3	Isian Form Identitas Aplikasi	56
Gambar 4.4	Isian Form Bobot Faktor Kualitas	57
Gambar 4.5	Langkah Pembobotan di Ms.Excel	59
Gambar 4.6	Isian Form Bobot Sub Faktor Kualitas	60
Gambar 4.7	Tampilan Bobot Faktor dan Sub Faktor Kualitas	62
Gambar 4.8	Isian Form Kriteria	63
Gambar 4.9	Rekomendasi Keseluruhan	65
Gambar 4.10	Rekomendasi Functionality	66
Gambar 4.11	Rekomendasi Reliability	66
Gambar 4.12	Rekomendasi Usability	67
Gambar 4.13	Rekomendasi Maintainability	67
Gambar 4.14	Rekomendasi Efficiency	68
Gambar 4.15	Rekomendasi Portability	68
Gambar 4.16	Menu dan Sub Menu Data	69
Gambar 4.17	Sub Menu Data Faktor Kualitas	69

Gambar 4.18	Sub Menu Data Sub Faktor Kualitas	70
Gambar 4.19	Sub Menu Data Kriteria	70
Gambar 4.20	Menu dan Sub Menu Edit	71
Gambar 4.21	Sub Menu Edit Faktor Kualitas	71
Gambar 4.22	Sub Menu Edit Sub Faktor Kualitas	72
Gambar 4.23	Sub Menu Edit Kriteria	72
Gambar 4.24	Menu dan Sub Menu Aplikasi	73
Gambar 4.25	Rekomendasi Global Skenario 2	73
Gambar 4.26	Rekomendasi Functionality Skenario 2	73
Gambar 4.27	Rekomendasi Global Skenario 3	74
Gambar 4.28	Rekomendasi Functionality Skenario 3	74
Gambar 4.29	Rekomendasi Global Skenario 4	75
Gambar 4.30	Rekomendasi Maintainability Skenario 4	76



## DAFTAR TABEL

Tabel 2.1	Perbandingan Model Kualitas	9
Tabel 2.2	Definisi Sub-Faktor Kualitas ISO 9126	17
Tabel 2.3	Urutan Faktor Kualitas dari Aplikasi Web	20
Tabel 2.4	Faktor Kualitas dari Aplikasi B2B	20
Tabel 2.5	Kriteria Kualitas Aplikasi B2B	21
Tabel 2.6	Faktor Kualitas vs Kriteria Kualitas	23
Tabel 2.7	Fungsi Keanggotaan Skala Linguistik	25
Tabel 2.8	Fungsi Keanggotaan Skala Linguistik Inversi	25
Tabel 3.1	Panduan Penilaian Kriteria	28
Tabel 3.2	Saran untuk Perbaikan Kualitas	33
Tabel 3.3	Pemetaan Kriteria ke Sub-Faktor Kualitas	34
Tabel 3.4	Keterangan Tabel	37
Tabel 3.5	Matrik Perbandingan Faktor Kualitas	43
Tabel 3.6	Matrik Perbandingan Sub-Faktor Kualitas	44
Tabel 3.7	Bobot Relatif dan Absolut	44
Tabel 3.8	Isian Faktor Kualitas	46
Tabel 3.9	Isian Sub-Faktor Kualitas	46
Tabel 3.10	Isian Kriteria	47
Tabel 3.11	Pemetaan Kriteria	47
Tabel 4.1	Isian Tabel users	55
Tabel 4.2	Isian Tabel aplikasi	57
Tabel 4.3	Skor Penilaian	57
Tabel 4.4	Isian Tabel matrik_fk	58
Tabel 4.5	Matrik Perbandingan Faktor Kualitas	58
Tabel 4.6	Konversi Faktor Kualitas ke Bilangan Fuzzy	58
Tabel 4.7	Isian Matrik Efficiency	60
Tabel 4.8	Isian Matrik Functionality	60
Tabel 4.9	Isian Matrik Maintainability	60
Tabel 4.10	Isian Matrik Portability	60
Tabel 4.11	Isian Matrik Reliability	61
Tabel 4.12	Isian Matrik Usability	61
Tabel 4.13	Hasil Bobot	61
Tabel 4.14	Isian Penilaian Kriteria	63
Tabel 4.15	Hasil Rekomendasi	64

# **BAB 1**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Perangkat lunak telah digunakan secara luas pada bermacam bidang aplikasi, dan sering menjadi titik kritis dari kesuksesan bisnis. Pembuatan atau pemilihan produk perangkat lunak berkualitas tinggi sangat penting. Spesifikasi yang komprehensif dan evaluasi kualitas produk perangkat lunak merupakan faktor kunci untuk menjamin kualitas.

Terdapat dua pendekatan inti yang dapat diikuti untuk menjamin kualitas produk. Pertama dengan penjaminan proses pengembangan produk, dan yang kedua dengan evaluasi kualitas produk akhir (Pressman, 2000). Untuk evaluasi kualitas produk akhir, dibutuhkan sekumpulan karakteristik kualitas yang mendeskripsikan produk dan membentuk dasar dari evaluasi (Fitzpatrick, 1996).

Sekumpulan karakteristik dan hubungan diantaranya adalah model kualitas (Botella, 2004) yang menyediakan dasar untuk spesifikasi kebutuhan kualitas dan kualitas evaluasi. Contoh penggunaan model kualitas adalah untuk validasi kelengkapan dari definisi kebutuhan, identifikasi kebutuhan perangkat lunak, identifikasi desain perangkat lunak dan sasaran pengujian serta identifikasi kriteria penerimaan untuk produk perangkat lunak yang lengkap.

Terdapat bermacam model kualitas, contohnya: FCM, McCall, Boehm, FURPS, Dromey, ISO 9126, SATC dan QMOOD. Model ISO lebih lengkap jika dibandingkan dengan model-model yang lain.

Pada penelitian sebelumnya (Behkamal, 2008), telah dilakukan kustomisasi ISO 9126 untuk evaluasi aplikasi B2B. Pada penelitian ini akan dikembangkan suatu model penilaian kualitas aplikasi web berupa CASE Tool dengan menggunakan ISO 9126 hasil kustomisasi tersebut. CASE Tool ini diharapkan dapat memudahkan pengguna dalam menilai kualitas aplikasi web secara lebih detail sekaligus dapat memberikan rekomendasi berupa saran-saran tindakan untuk perbaikan kualitas.

## **1.2 Perumusan Masalah**

Permasalahan yang dihadapi dalam penelitian ini adalah:

1. Bagaimana melakukan pembobotan menggunakan *Fuzzy Analytical Hierarcycal Process* (FAHP) untuk menilai kualitas aplikasi web.
2. Bagaimana memandu pengguna dalam melakukan penilaian terhadap kriteria kualitas
3. Bagaimana membuat CASE tool yang dapat menilai kualitas aplikasi web dan memberikan rekomendasi untuk perbaikan kualitas.

## **1.3 Tujuan dan Manfaat Penelitian**

Tujuan dari penelitian adalah membuat suatu model penilaian kualitas aplikasi web berdasarkan ISO 9126.

Manfaat dari penelitian adalah memungkinkan pihak pengembang aplikasi web memperoleh penilaian kualitas secara lebih komprehensif.

Adapun kontribusi dari penelitian adalah:

1. Adanya suatu CASE tool yang selain dapat menilai kualitas aplikasi web juga dapat memberikan rekomendasi perbaikan kualitas.
2. Adanya hasil pembobotan karakteristik kualitas aplikasi web dengan menggunakan FAHP.
3. Adanya pemetaan kriteria kualitas aplikasi ke sub faktor kualitas dari ISO 9126.

## **BAB 2**

### **KAJIAN PUSTAKA DAN DASAR TEORI**

Pada bab ini akan dibahas tentang berbagai tinjauan pustaka yang dipakai penulis dalam pembuatan Penelitian ini. Bab ini membahas kualitas perangkat lunak, model kualitas, ISO 9126, aplikasi web dan Fuzzy AHP.

#### **2.1 Kualitas Perangkat Lunak**

Berbagai macam definisi kualitas perangkat lunak (*software quality*) bergantung pada darimana pemakai memandang dan melihat sesuai dengan kebutuhannya. (Crosby, 1979) mendefinisikan kualitas atau mutu sebagai "*conformance to requirements*". Selama seseorang dapat berdebat tentang perbedaan antara kebutuhan, keinginan dan kemauannya, definisi kualitas harus mempertimbangkan perspektif pemakai tersebut. Kunci utama pertanyaan untuk sebuah definisi kualitas adalah siapa pemakainya, apa yang penting bagi mereka dan bagaimana prioritasnya tentang metode apa yang dibangun, dibungkus untuk mendukung sebuah produk.

Untuk menjawab pertanyaan tersebut, harus dikenali hirarki dari kualitas perangkat lunak. Pertama, suatu produk perangkat lunak harus menyediakan fungsi suatu jenis dan waktu yang sama ketika pemakai memerlukannya. Kedua, produk harus berjalan. Jika produk memiliki kecacatan maka produk tersebut tentunya tidak ada konsistensi kelayakan. Para pemakai tidak akan menggunakannya dengan mengabaikan atribut-atribut yang menyertainya. Hal tersebut tidak berarti bahwa kecacatan selalu menjadi prioritas yang paling utama dalam menolak suatu produk, tetapi akan menjadi sangat penting dalam melihat layak atau tidaknya. Jika tingkatan cacat minimum belum dicapai maka berbagai hal tidak ada yang perlu dipertimbangkan. Di luar ambang kualitas tersebut, bagaimanapun juga sesuatu yang berhubungan dengan pertimbangan dan penilaian cacat suatu produk perangkat lunak seperti halnya kegunaan, kecocokan, kemampuan, dan lainnya tergantung pada pemakai tersebut memandang dan menilainya termasuk didalamnya aplikasinya dan lingkungan perangkat lunak yang menyertainya (Humphrey, 1994).

Dalam hal kualitas, The Institute of Electrical and Electronics Engineers (IEEE, 1990) Standard Glossary of Software Engineering Terminology mendefinisikan kualitas sebagai: “Tingkat dimana sebuah sistem, komponen atau proses dapat memenuhi spesifikasi kebutuhannya dan kebutuhan/keinginan/ harapan dari pengguna”. Definisi dari IEEE digunakan dalam konteks suatu sistem perangkat lunak secara rinci. Dimana kualitas didefinisikan sebagai tingkat atau level bagaimana sebuah sistem, komponen atau proses memenuhi kebutuhan yang diminta pengguna atau harapan pengguna. Kualitas adalah suatu atribut dari sistem yang berjalan, dan sangat erat kaitannya dengan resiko. Semakin tinggi resiko yang didapatkan dan kemudian dikurangi maka akan tinggi kualitas yang dihasilkannya. Dengan cara yang sama, lebih cepat resiko dikenali dan dikurangi, akan lebih tinggi pula kualitasnya.

Menurut (Pressman, 2000) kualitas perangkat lunak adalah konformansi terhadap kebutuhan fungsional dan kinerja yang dinyatakan secara eksplisit, standar perkembangan yang didokumentasikan secara eksplisit dan karakteristik implisit yang diharapkan bagi semua perangkat lunak yang dikembangkan secara profesional. Kualitas perangkat lunak didefinisikan sebagai kesesuaian yang diharapkan pada semua perangkat lunak yang dibangun dengan mengutamakan fungsi, unjuk kerja, standar pembangunan yang terdokumentasi dan karakteristik yang ditunjukkannya. Definisi ini menekankan pada 3 hal yaitu:

1. Kebutuhan perangkat lunak adalah fondasi ukuran kualitas, jika suatu perangkat lunak tidak sesuai dengan kebutuhan yang ditentukan maka kualitasnya berkurang.
2. Jika menggunakan suatu standar untuk pembangunan perangkat lunak, maka ketika tidak memenuhi standar tersebut maka dianggap kurang berkualitas
3. Seringkali ada kualitas yang secara langsung diutarakan (tersirat) seperti kemudahan penggunaan dan pemeliharaan yang baik. Kualitas perangkat lunak dipertanyakan jika tidak memenuhi kebutuhan ini.

Menurut ISO 8204, kualitas perangkat lunak adalah: “Keseluruhan karakteristik dari suatu kesatuan dalam kemampuannya untuk memenuhi dan memuaskan pemakai yang dinyatakan dan disiratkan dalam suatu kebutuhan.” Artinya, diperlukan suatu kualitas produk perangkat lunak yang mempunyai

karakteristik tertentu yang dihubungkan dengan kebutuhan pemakai dan membuat puas penggunaannya. ISO menyoroti pada fitur-fitur dan karakteristik dari produk atau layanan dalam kemampuannya memenuhi kebutuhan yang ditentukan. ISO menyediakan model yang berbasiskan obyek dalam 3 konteks dasar yaitu: kualitas, kebutuhan dan karakteristik. Standar dapat membantu mendefinisikan suatu terminologi, seperti halnya kata “kualitas” (*quality*).

Kebutuhan dan karakteristik berperan penting dalam mendefinisikan suatu kualitas. Oleh karena itu, suatu model yang berbasiskan obyek bermanfaat dalam pemahaman yang lebih baik untuk masalah ini. Gambar 2.1 menunjukkan suatu produk perangkat lunak, dimana untuk memenuhi suatu kebutuhan diperlukan karakteristik yang sesuai. Keberadaan hubungan antara kebutuhan dan karakteristik menjadikan dimungkinkannya pernyataan yang jelas tentang kualitas suatu produk.



Gambar 2.1 Hubungan Kebutuhan dan Karakteristik Perangkat Lunak

Adapun definisi kualitas perangkat lunak dalam the HandBook of Software Quality Assurance dalam berbagai makna, mengerucut pada definisi : “*Software quality is the fitness for use of the software product*” (Schulmeyer, 1998).

Kualitas perangkat lunak adalah keberadaan karakteristik dari suatu produk yang dijabarkan dalam kebutuhannya, artinya harus dilihat terlebih dahulu karakteristik-karakteristik apa yang berhubungan atau tidak dengan kebutuhan-kebutuhan yang diinginkan oleh pemakai. Karakteristik yang dimaksud yaitu karakteristik netral dan kontra-produktif. Pemahaman tentang karakteristik tersebut diperlukan untuk mengurangi kontra produktif dari kualitas perangkat lunak yang dimaksud serta relevan atau tidaknya perangkat lunak tersebut untuk kebutuhan suatu organisasi. Tidak hanya adanya keberadaan karakteristik tersebut tetapi juga tidak

adanya kontra produktif dari karakteristik suatu perangkat lunak yang diinginkan (Petrascch, 1999).

Pertanyaan pertama yang muncul ketika membahas pengukuran kualitas perangkat lunak, adalah apa yang sebenarnya mau diukur. Kualitas perangkat lunak dapat dilihat dari sudut pandang proses pengembangan perangkat lunak dan hasil produk yang dihasilkan. Dan penilaian ini tentu berorientasi akhir ke bagaimana suatu perangkat lunak dapat dikembangkan sesuai dengan yang diharapkan oleh pengguna.

Dari sudut pandang produk, parameter dan metode pengukuran menurut (Wahono, 2006), *“When you can measure what you are speaking about, and express it in numbers, you know something about it. But when you can not measure it, when you can not express it in numbers, your knowledge is of a meagre and unsatisfactory kind”*.

Kualitas perangkat lunak tidak dapat dikhususkan hanya pada perangkat lunak yang tidak memiliki kesalahan. Pendekatan teknik menginginkan agar kualitas perangkat lunak dapat diukur secara kuantitatif, dalam bentuk angka-angka yang mudah dipahami oleh manusia. Spesifikasi kualitas perangkat lunak harus lebih akurat dan detail, karenanya perlu ditentukan parameter atau atribut pengukuran. Formalisasi dari kualitas perangkat lunak dapat dilakukan dengan menggunakan model kualitas.

## **2.2 Model Kualitas Perangkat Lunak**

Model kualitas merepresentasikan sebuah interaksi antara sekelompok karakteristik dan sub-karakteristik yang berfungsi sebagai dasar spesifikasi kebutuhan kualitas untuk penilaian kualitas (Khosravi, 2005). Model kualitas dibutuhkan untuk karakterisasi kualitas produk perangkat lunak. Terdapat sejumlah besar propertis dari perangkat lunak yang nyata dan diketahui berpengaruh positif terhadap kualitas. Propertis tersebut belum pernah diorganisasi ke dalam framework sistematis yang dapat memaksimalkan pengaruh dari bobot kumulatif. Karenanya dibutuhkan perlakuan yang tepat untuk membangun model praktis dari kualitas produk perangkat lunak. Beberapa propertis perangkat lunak disebut kualitas atau atribut kualitas.

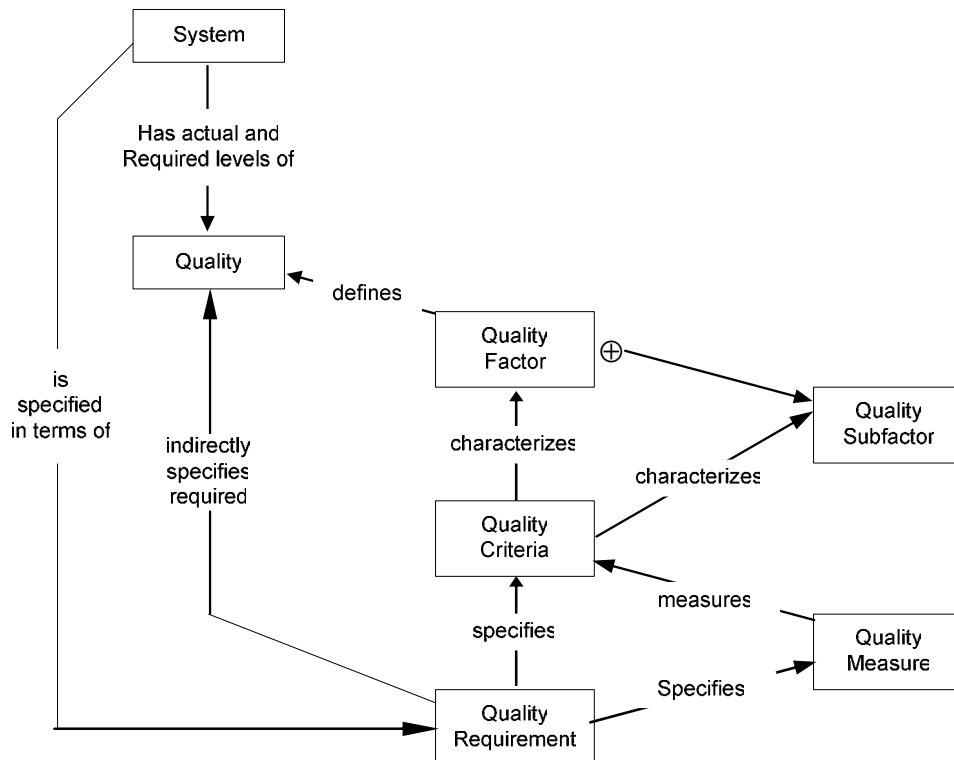
Model kualitas adalah sebuah konsep yang berasal dari kualitas dan pengukuran. Model kualitas menurunkan konsep umum kualitas untuk membangun sebuah taksonomi dari komponen faktor dan subfaktor kualitas (aspek, atribut atau karakteristik). Selanjutnya menyediakan kriteria kualitas dan pengukuran khusus yang dapat digunakan untuk membagi faktor kualitas level tinggi secara umum menjadi pengukuran yang spesifik dan detail. Sehingga dapat digunakan untuk spesifikasi aspek kualitas yang berhubungan atau menentukan selama testing, apakah terdapat aspek dari kualitas. Cakupan dari model kualitas menurut (Khosravi, 2004):

- Satu atau lebih aplikasi yang berhubungan.
- Satu atau lebih komponen yang berhubungan.
- Model kualitas standar dari industri secara umum, contoh ISO 9126, OPF
- Model kualitas sebuah organisasi, contoh semua aplikasi pada program atau proyek yang berhubungan atau *product-line* dari aplikasi.
- Model kualitas *endeavor-specific*, seperti yang digunakan pada proyek tunggal yang membangun aplikasi tunggal.

Sehingga, model kualitas dapat digunakan untuk dokumentasi atau analisa kualitas unit bisnis yang aktual atau terencana (Firesmith, 2003). Definisi komponen kualitas model yang ditunjukkan pada Gambar 2.2 adalah:

- *Quality Factor Groups*: kumpulan faktor kualitas yang berhubungan
- *Quality Factors*: aspek, atribut atau karakteristik level tinggi dari sebuah aplikasi, komponen atau organisasi bisnis.
- *Quality Subfactors*: faktor kualitas level lebih rendah yang merupakan komponen dari faktor atau sub-faktor yang lain
- *Quality Criteria*: deskripsi spesifik yang menyediakan evidence atau against eksistensi dari faktor atau sub-faktor kualitas yang spesifik.
- *Quality Measures*: metrik yang menentukan kriteria kualitas dan membuatnya dapat diukur, objektif dan tidak ambigu.





Gambar 2.2 Hubungan Komponen Model Kualitas (Firesmith, 2003)

Kualitas, atau lebih spesifiknya, sekumpulan atribut kualitas adalah sarana bagi *interest group* yang berbeda untuk menyatakan kebutuhannya akan perangkat lunak. Pembuatan model kualitas untuk perangkat lunak, efektif untuk mengakomodasi dan menyeimbangkan kebutuhan dari *interest group* yang berbeda. Sekumpulan propertis atau atribut kualitas dari perangkat lunak menyediakan abstraksi atau spesifikasi level tinggi untuk kualitas produk perangkat lunak.

Pengukuran perangkat lunak merupakan hal yang sangat penting pada teknik perangkat lunak, maka metrik digunakan untuk mengevaluasi kuantitas dan kualitas model. Pemilihan metode sangat dipengaruhi oleh karakteristik dari aplikasi yang akan dianalisa.

Model kualitas dibutuhkan untuk karakterisasi kualitas produk perangkat lunak. Terdapat sejumlah besar propertis dari perangkat lunak yang nyata dan diketahui berpengaruh positif terhadap kualitas. Propertis tersebut belum pernah diorganisasi ke dalam framework yang sistematis yang dapat memaksimalkan pengaruh dari bobot kumulatif. Karenanya dibutuhkan pembangunan model praktis

dari kualitas produk perangkat lunak. Beberapa propertis perangkat lunak disebut kualitas atau atribut kualitas.

Kualitas, atau lebih spesifiknya, sekumpulan atribut kualitas adalah sarana bagi interest group yang berbeda untuk menyatakan kebutuhannya akan perangkat lunak. Pembuatan model kualitas untuk perangkat lunak, efektif untuk mengakomodasi dan menyeimbangkan kebutuhan dari interest group yang berbeda. Sekumpulan propertis atau atribut kualitas dari perangkat lunak menyediakan abstraksi atau spesifikasi level tinggi untuk kualitas produk perangkat lunak.

Menurut taksonomi McCall, atribut tersusun secara hirarkis, dimana level atas (*high-level attribute*) disebut faktor (*factor*), dan level bawah (*low-level attribute*) disebut dengan kriteria (*criteria*). Faktor menunjukkan atribut kualitas produk dilihat dari sudut pandang pengguna. Sedangkan kriteria adalah parameter kualitas produk dilihat dari sudut pandang perangkat lunaknya sendiri. Faktor dan kriteria ini memiliki hubungan sebab akibat (*cause-effect*).

Hubungan antara kualitas perangkat lunak secara hierarki ataupun secara non hierarki adalah analisa faktor-faktor perangkat lunak untuk mengevaluasi aplikasi, dalam penelitian ini hanya dibahas perbandingan dengan struktur hierarki. Pemilihan metode sangat dipengaruhi oleh karakteristik dari aplikasi yang akan dianalisa.

Pada Tabel 2.1 akan dianalisa lima model kualitas yang berstruktur hirarki untuk membuktikan bahwa ISO 9126 merupakan model yang terlengkap.

Tabel 2.1 Perbandingan model kualitas

Istilah	Definisi	Kriteria pengujian
Mc Call (McCall, 1977)		
Maintainability	Berusaha menemukan dan memperbaiki kesalahan dari program	simplicity, conciseness, instrumentation, self descriptiveness
Flexibility	Memodifikasi operational dari program	simplicity, expandability, generality, modularity
Testability	Melakukan test pada program untuk memastikan fungsi program	simplicity, instrumentation, self descriptiveness, modularity
Portability	Transfer program dari hardware atau software ke lainnya	simplicity, software system independence, machine independence
Reusability	Menentukan apakah program dapat digunakan kembali oleh aplikasi lain	simplicity, generality, software system independence, machine indepenence
Interoperability	Menghubungkan satu sistem	modularity, communications commonality, data

	pada sistem lain	communality
Correctness	Pengukuran keberhasilan sistem diperuntukkan untuk setiap kelompok konsumen	traceability, completeness, consistency
Reliability	Program dapat memberikan fungsi dan hasil yang tepat	consistency, accuracy, error tolerance
Efficiency	Kebutuhan resources dan code dari program untuk menghasilkan fungsi	execution efficiency, storage efficiency
Integrity	Mengakses software atau data oleh <i>unauthorized people</i> dapat dikontrol	access control, access audit
Usability	Pembelajaran, pengoperasian, persiapan input dan hasil dari program	Operability, training, communicativeness
Boehm (Botella, 2004)		
Portability	Menentukan software bekerja dengan perbedaan konfigurasi komputer misalkan sistem operasi, databases.	device independence, self containedness
Reliability	Menentukan penampilan software sesuai yang dibutuhkan, misalkan jumlah dari kekurangan/defect.	self containedness, Accuracy, Completeness, Robustness/Integrity, Concistency
Efficiency	Penggunaan optimum dari sistem resources selama eksekusi.	Accountability, Device Efficiency, Accessibility
Usability/Human engineering	Mudah dalam penggunaan	robustness/integrity, accessibility, communicativeness
Testability	Kemudahan dalam validasi, kebutuhan dari software	Accountability, Accessibility, Communicativeness, Self descriptiveness, Structuredness
Understandability	Menentukan software yang mana yang mudah dipahami sesuai tujuan dan struktur.	Consistency, self descriptiveness, structuredness, conciseness, legibility
Flexibility/Modifiability	Mudah dalam perubahan software sesuai dengan kebutuhan.	Structuredness, augmentability
FURPS (Grady, 1992)		
Functionality	Kumpulan dari feature dan kemampuan dari program, fungsi yang bersifat general dan pengamanan dari beberapa sistem	feature sets, capabilities and security
Usability	Faktor manusia, estetika, konsistensi pada interface dan dokumentasi.	human factors, aesthetics, consistency in the user interface, online context-sensitive help, wizard ,agents, user documentation,training materials
Reliability	Frekuensi kesalahan, kebenaran dari hasil output, waktu antara kesalahan, kemampuan merecover dari kesalahan, dan kemampuan prediksi program.	frequency failure, severity of failure, recoverability, predictability, accuracy, mean time between failure (MTBF)
Performance	Kecepatan dalam memproses, waktu untuk meresponse, konsumsi resource, efisiensi	speed, efficiency, availability, accuracy, throughput, response time, recovery time, resource usage
Supportability	Kemampuan memperluas	testability, extensibility, adaptability,

	program, adaptasi, serviceability, uji coba, configureability, penentuan system dapat diinstal dimana, penentuan permasalahan ditentukan dalam permasalahan apa.	maintainability, compatibility, configurability, serviceability, installability, localizability (internationalization)
ISO 9126 (ISO/IEC 9126-1)		
Functionality	Kemampuan menutupi fungsi produk perangkat lunak yang menyediakan kepuasan kebutuhan user.	Access audit ability, Access controllability, Audit ability, Clarity, Communication Commonality, Completeness, Computational Accuracy, Correctness, Data Commonality, Uniformity.
Reliability	Kemampuan perangkat lunak untuk perawatan dengan level performansi.	Degradability, Failure avoidance, Incorrect operation avoidance, Instrumentation, Mean recovery time, Restart ability, Restorability, Robustness, Trust ability
Usability	Kemampuan yang berhubungan dengan penggunaan perangkat lunak	Access audit ability, Clarity, Completeness of description, Demonstration accessibility, Effectiveness of help System in use, Effectiveness of the user documentation, Globalization, Simplicity, Uniformity
Efficiency	Kemampuan yang berhubungan dengan sumber daya fisik yang digunakan ketika perangkat lunak dijalankan	I/O devices Utilization, Memory utilization, Response time, Throughput, Turnaround Time
Maintainanility	Kemampuan yang dibutuhkan untuk membuat perubahan perangkat lunak	Audit ability, Audit trail capability, Completeness of description, Complexity, Modification complexity, Modularity, Reusability, Self-Description Documentation
Portability	Kemampuan yang berhubungan dengan kemampuan perangkat lunak yang dikirim ke lingkungan berbeda	Ease of installation, Expendability, Hardware environmental adaptability, Hardware Independence, Organizational environment adaptability, Software Independence, Withdraw ability,
Suitability	Fungsi dari aplikasi cocok dengan kebutuhan user untuk memenuhi tugas dengan meminimumkan keikutsertaan user.	Clarity, Communication Commonality, Data Commonality, Uniformity.
Accuracy	Analisa dari hasil aplikasi benar	Access audit ability, Access controllability, Audit ability, Clarity, Completeness, Computational Accuracy, Correctness.
Interoperability	Aplikasi dapat melakukan interaksi dengan <i>specified system</i>	Access audit ability, Access controllability, Communication Commonality, Data Commonality, Uniformity.
Security	Kemampuan untuk mencegah <i>unauthorized access</i> secara accidental /deliberate pada program dan data.	Access audit ability, Access controllability, Audit ability
Standart/ compliance	Aplikasi sesuai dengan standart quality, persamaan persepsi/pendapat dan persamaan dalam hukum.	Data Commonality, Uniformity.
Maturity	Frekuensi kesalahan dari software	Degradability, Incorrect operation avoidance, Instrumentation
Fault Tolerance	Kemampuan dari software dalam	Failure avoidance, Incorrect operation

	hal menyesuaikan toleransi kesalahan dari software secara spesifik.	avoidance, Instrumentation, Robustness, Trust ability
Recoverability	Merupakan kemampuan untuk merecover data yang terpengaruh oleh kesalahan dan pengukuran waktu dan usaha dalam merecover.	Instrumentation, Mean recovery time, Restart ability, Restorability, Robustness, Trust ability
Understandability	Mendeskripsikan usaha dari user untuk mengenali konsep logika dari aplikasi dan kemampuan dari logika konsep.	Clarity, Completeness of description, Demonstration accessibility, Effectiveness of help System in use, Effectiveness of the user documentation, Globalization, Simplicity, Uniformity
Lernability	Usaha dari user dalam mempelajari aplikasi contoh : kontrol operasi, input dan output dan hasil akhir	Access audit ability, Clarity, Completeness of description, Demonstration accessibility, Effectiveness of help System in use, Effectiveness of the user documentation, Globalization, Simplicity, Uniformity
Operability	Usaha user untuk melakukan operasi dan mengontrol operasi	Access audit ability, Clarity, Completeness of description, Demonstration accessibility, Effectiveness of help System in use, Effectiveness of the user documentation, Globalization, Simplicity, Uniformity
Timebased Efficiency	Waktu penggunaan	I/O devices Utilization, Response time, Throughput, Turnaround Time
Resourcebased Efficiency	Jumlah resources yang digunakan.	I/O devices Utilization, Memory utilization, Throughput, Turnaround Time
Analysability	Usaha yang dibutuhkan untuk mendiagnosa dalam mempertahankan, kesalahan, dan identifikasi bagian yang akan dimodifikasi.	Audit ability, Audit trail capability, Completeness of description, Complexity, Modification complexity, Self-Description Documentation
Changeability	Usaha yang dibutuhkan untuk memodifikasi, penghapusan kesalahandan perubahan lingkungan.	Complexity, Modification complexity, Modularity, Reusability
Stability	Merupakan toleransi dari aplikas jika terdapat efek yang tidak diharapkan dari modifikasi.	Completeness of description, Complexity, Modification complexity, Modularity, Reusability, Self-Description Documentation
Testability	Merupakan usaha untuk memvalidasi modifikasi.	Completeness of description, Complexity, Modification complexity, Modularity, Self-Description Documentation
Adaptability	Merupakan kesempatan untuk mengadaptasi aplikasi pada lingkungan yang berbeda tanpa pertambahan usaha	Ease of installation, Expendability, Hardware environmental adaptability, Hardware Independence, Organizational environment adaptability, Software Independence, Withdraw ability
Installability	Merupakan usaha untuk melakukan instalasi software	Ease of installation, Expendability, Hardware environmental adaptability, Hardware Independence, Organizational environment adaptability, Software Independence, Withdraw ability
Conformance	Merupakan penyesuaian diri dari aplikasi untuk standard atau berhubungan dengan portability	Ease of installation, Expendability, Hardware environmental adaptability, Hardware Independence, Organizational environment adaptability, Software Independence, Withdraw ability

Replaceability	Merupakan kesempatan dan usaha untuk menggunakan aplikasi sebagai pengganti dari aplikasi lain.	Ease of installation, Expendability, Hardware environmental adaptability, Hardware Independence, Organizational environment adaptability, Software Independence, Withdraw ability
Dromey (Dromey, 1995)		
Correctness	Melakukan seperti apa yang direncanakan	Functionality, reliability
Internal	Kebenaran internal misalkan kebiasaan dari loop	Maintainability, efficiency, reliability
Contextual	Kualitas relational	Maintainability, reusability, portability, reliability
Descriptive	Mendeskripsikan proses pada semua stage	Maintainability, reusability, portability, usability

## 2.3 ISO 9126

International Organization untuk Standardisation (ISO) didirikan pada 1946 dalam upaya untuk memfasilitasi perdagangan internasional, koordinasi internasional dan penyamaan standar dengan memberikan satu definisi standar yang akan digunakan dan diakui. Setelah beberapa tahun, beberapa daftar karakteristik Kualitas perangkat lunak ditampilkan. Mengetahui kesulitan pada definisi kualitas perangkat lunak yang baik dengan cara, misalnya menjadikan kesenangan kepada kesalahan perangkat lunak yang dapat ditolerir dan diperbaiki. Untuk beberapa ‘ketahanan’ (robustness) yang berarti toleransi kesalahan input pada perangkat lunak, dengan kemampuan untuk merubah kode program tanpa menampilkan kesalahan. ISO 9126 dibuat pada 1991 untuk memberikan sebuah framework untuk mengevaluasi kualitas software dan kemudian diperbaharui selama periode sepuluh tahun melalui pertanyaan tentang definisi Kualitas perangkat lunak. Banyak penelitian mengkritik ISO 9126 karena tidak menjelaskan secara spesifik kebutuhan kualitas, tetapi hanya mendefinisikan framework secara umum untuk mengevaluasi kualitas software.

ISO 9126 merupakan salah satu model kualitas yang telah distandarisasi secara internasional. Perbedaan dengan model kualitas yang lain terdapat pada hubungan antar faktor kualitas dimana kalau ISO mempunyai hubungan 1:N, dimana setiap karakteristik memiliki subkarakteristiknya sendiri dan tidak berhubungan dengan satu atau lebih karakteristik yang lainnya. Dalam ISO 9126, yang menjadi karakteristik untuk kualitas perangkat lunak ada pada Tabel 2.1.

Standard ISO 9126 sangat panjang (ISO, 9126) dikarenakan orang memiliki motivasi berbeda yang memungkinkan untuk tertarik pada kualitas perangkat lunak :

- *Acquirer*: orang yang memperoleh perangkat lunak dari supplier eksternal.
- *Developer*: orang yang membangun produk perangkat lunak.
- *Evaluator independent*: orang yang menetapkan kualitas produk perangkat lunak – tidak untuk dirinya sendiri tetapi untuk komunitas pengguna – misalnya melalui jenis tool tertentu dari sebuah perangkat lunak sebagai bagian dari aktifitas profesional.

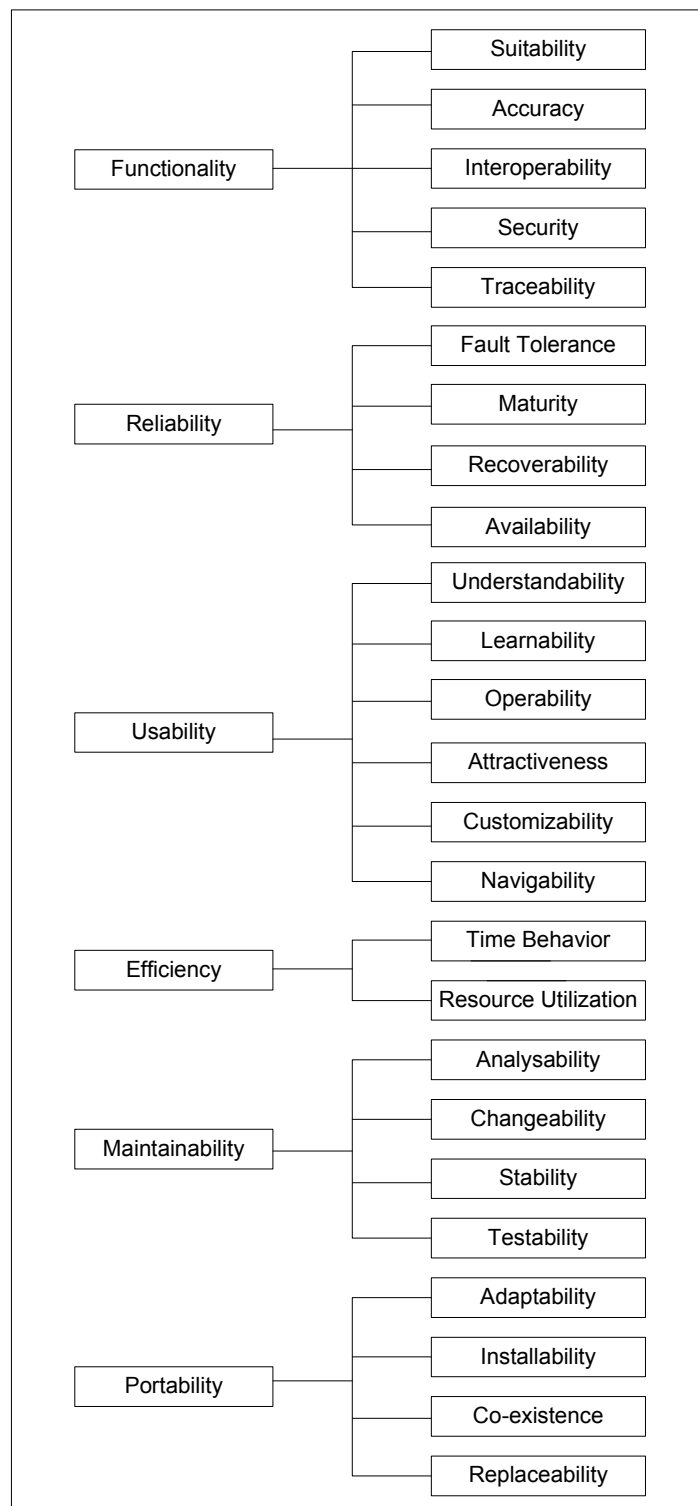
ISO 9126 telah membagi dokumen menjadi tiga bagian kebutuhan. Disamping ukuran bagian dokumentasi, ISO 9126 tidak hanya mendefinisikan atribut kualitas perangkat lunak. Standard ISO 14598 memisahkan prosedur yang seharusnya dibawa saat menaksir derajat produk perangkat lunak untuk menyesuaikan diri pada karakteristik kualitas ISO 9126 yang dipilih. Hal ini mungkin saja tidak diperlukan, tetapi disetujuinya ISO 14598 dapat digunakan untuk menyelesaikan penilaian dalam membedakan bagian karakteristik kualitas pada ISO 9126 yang dibutuhkan.

Perbedaan antara atribut kualitas internal dan eksternal telah dicatat, ISO 9126 juga memperkenalkan tipe kualitas – *quality in use* – dimana mengikuti elemen yang telah diketahui :

- *Effectiveness*: kemampuan untuk mencapai tujuan pengguna melalui akurasi dan kelengkapan.
- *Productivity*: upaya menghindari kelebihan penggunaan sumber daya, seperti biaya staf dalam mencapai tujuan pengguna.
- *Safety*: upaya menghindari kejahatan level resiko untuk orang dan entitas lain seperti bisnis, perangkat lunak, properti dan lingkungan
- *Satisfaction*: kepuasan pengguna dalam menggunakan perangkat lunak.

Pengguna pada konteks ini adalah orang yang tidak hanya bekerja secara nyata pada sistem perangkat lunak yang akan dibuat, tetapi juga orang yang akan merawat dan meningkatkan perangkat lunak. Ide kualitas dalam penggunaan underlines adalah bagaimana mempersiapkan kualitas perangkat lunak sebagai

atribut yang tidak hanya berlaku pada perangkat lunak tetapi juga pada konteks penggunaan.



Gambar 2.3 Model Kualitas ISO 9126 (Behkamal, 2008)



ISO 9126 mengidentifikasi enam karakteristik kualitas perangkat lunak utama seperti pada Gambar 2.3 yaitu:

- *Functionality*: kemampuan menutupi fungsi produk perangkat lunak yang menyediakan kepuasan kebutuhan pengguna.
- *Reliability*: kemampuan perangkat lunak untuk perawatan dengan level performansi.
- *Usability*: kemampuan yang berhubungan dengan penggunaan perangkat lunak.
- *Efficiency*: kemampuan yang berhubungan dengan sumber daya fisik yang digunakan ketika perangkat lunak dijalankan.
- *Maintainability*: kemampuan yang dibutuhkan untuk membuat perubahan perangkat lunak
- *Portability*: kemampuan yang berhubungan dengan kemampuan perangkat lunak yang dikirim ke lingkungan berbeda.

ISO 9126 menyarankan sub-karakteristik untuk setiap karakteristik utama.

*Interoperability* merupakan gambaran yang bagus pada usaha ISO 9126 untuk mengklarifikasi terminologi *interopability* yang mengacu pada kemampuan perangkat lunak untuk berinteraksi dengan sistem lain. Kerangka ISO 9126 telah dipilih pada kata ini daripada *comparability* karena nantinya akan mengakibatkan kebingungan dengan karakteristik yang dituju oleh ISO 9126 sebagai *replaceability*.

*Maturity* mengacu pada frekuensi kesalahan produk perangkat lunak yang memberikan dampak pada perangkat lunak yang digunakan sehingga kesalahan menjadi tidak nampak dan mudah dihilangkan. Hal ini menarik untuk ketahui bahwa *recoverability* telah menjadi hal yang berbeda dari *security* yang menggambarkan kontrol akses pada sistem.

Tool perangkat lunak dapat dengan mudah dipelajari tetapi menghabiskan waktu untuk menggunakannya dikarenakan oleh cara penggunaannya membutuhkan jumlah menu besar. Hal ini dapat diaplikasikan untuk paket yang singkat, tetapi tidak ada sistem yang menggunakannya untuk sepanjang waktu setiap hari. Pada kasus ini *learnability* telah disatukan pada biaya *operability*. *Attractiveness* adalah tambahan terbaru pada sub-karakteristik *usability* dan sangat penting dimana pengguna tidak

dipaksa untuk menggunakan produk perangkat lunak tertentu. Misalnya pada kasus game dan produk entertainment lain.

*Analysability* merupakan kemudahan untuk menentukan penyebab kesalahan. *Changeability* merupakan kualitas lain dari *flexibility* yang kemungkinan nantinya disebut sebagai *changeability* yang memiliki arti tambahan yang sederhana dalam perbendaharaan kata bahasa Inggris. Hal ini mungkin saja menandakan pemasok perangkat lunak yang selalu berubah. Di sisi lain pengertian *stability*, adalah tidak berarti perangkat lunak itu tidak pernah berubah. Hal ini berarti juga terdapat resiko yang kecil pada modifikasi perangkat lunak yang memiliki dampak tidak diduga.

*Portability compliance* berhubungan erat dengan standard kemampuan yang digunakan pada platform manapun (*portability*). Standard bahasa pemrograman umum pada kasus ini digunakan pada lingkungan perangkat keras atau perangkat lunak. *Replaceability* mengarah ke faktor yang memberikan *upward compatibility* antara komponen software lama dan yang baru. Sedangkan *downwards compatibility* tidak dijelaskan pada pengertian definisi. *Co-existence* mengarah pada kemampuan software untuk berbagi sumber daya dengan komponen perangkat lunak lain, tetapi tidak seperti *interoperability*, dimana tidak ada data yang digunakan. Definisi sub-faktor kualitas dari ISO 9126 dirangkum dan dijelaskan pada Tabel 2.2.

Tabel 2.2 Definisi Sub-Faktor Kualitas ISO 9126

Sub-Faktor Kualitas	Definisi
Suitability	Gambaran perangkat lunak dalam menjalankan fungsi-fungsi yang ada pada perangkat lunak tersebut
Accuracy	Hasil yang diekspektasikan dari perangkat lunak
Fault Tolerance	Kemampuan perangkat lunak untuk menyesuaikan toleransi kesalahannya secara spesifik
Maturity	Frekuensi kesalahan produk perangkat lunak yang memberikan dampak pada perangkat lunak yang digunakan sehingga kesalahan menjadi tidak nampak dan mudah dihilangkan
Interoperability	Kemampuan perangkat lunak untuk berinteraksi dengan sistem lain
Security	Memperlihatkan bagaimana perangkat lunak melindungi dari akses yang belum diautorisasi
Recoverability	Kemampuan perangkat lunak untuk menetapkan kembali level performansinya dan menyelamatkan data apabila terjadi kesalahan dan kejadian tak terduga
Traceability	Kemampuan perangkat lunak untuk mencari kebenaran dari pemrosesan informasi pada tiap tahap proses yang berbeda
Understandability	Gambaran bagaimana user secara komprehensif menggunakan perangkat lunak dengan mudah
Learnability	Kemudahan perangkat lunak untuk dipelajari

Operability	Kemudahan perangkat lunak untuk dioperasikan oleh pengguna
Time Behavior	Menggambarkan seberapa cepat respon dari sistem
Resource Utilization	Menggambarkan seberapa efisien perangkat lunak dalam menggunakan sumber daya yang ada
Analysability	Kemudahan untuk menentukan penyebab kesalahan
Changeability	Kemampuan untuk memodifikasi dan menghapus kesalahan dalam perubahan lingkungan
Stability	Merupakan toleransi dari aplikasi jika terdapat efek yang tidak diharapkan dari modifikasi
Testability	Kemudahan perangkat lunak dalam mengadakan uji coba terhadapnya
Adaptability	Kemampuan perangkat lunak untuk beradaptasi ketika dipindahkan ke lingkungan yang berbeda
Installability	Kemudahan perangkat lunak untuk di install
Co-existence	Kemampuan perangkat lunak untuk berbagi sumber daya dengan komponen perangkat lunak lain
Replaceability	Faktor yang memberikan 'upward compatibility' antara komponen perangkat lunak lama dan yang baru
Availability	Keberadaan perangkat lunak bagi pengguna kapanpun dibutuhkan
Attractiveness	Tidak adanya paksaan bagi user untuk menggunakan produk perangkat lunak tertentu
Customizability	Kemampuan perangkat lunak untuk disesuaikan dengan kebutuhan pengguna dan meningkatkan kepuasan pengguna ketika menggunakan perangkat lunak
Navigability	Kemudahan bagi pengguna untuk mengakses informasi secara cepat dan efisien

ISO 9126 menyediakan petunjuk untuk menggunakan kualitas karakteristik. Variasi dalam membedakan karakteristik kualitas tergantung pada tipe produk yang ditekankan.

## 2.4 Aplikasi Web

Dalam rekayasa perangkat lunak, suatu aplikasi web (bahasa Inggris: *web application* atau sering disingkat *webapp*) adalah suatu aplikasi yang diakses menggunakan penjelajah web melalui suatu jaringan seperti internet atau intranet. *Webapp* juga merupakan suatu aplikasi perangkat lunak komputer yang dikodekan dalam bahasa yang didukung penjelajah web (seperti HTML, JavaScript, AJAX, Java, dll) dan bergantung pada penjelajah tersebut untuk menampilkan aplikasi.

Pada form paling sederhana, dapat lebih kecil daripada link file *hypertext* yang memberikan informasi dengan menggunakan teks dan grafis terbatas. Sebagaimana aplikasi e-commerce dan B2B yang semakin penting sekarang ini, aplikasi web berkembang menjadi lingkungan komputasi yang sangat menarik yang

tidak hanya menyediakan fitur *standalone*, fungsi komputing dan konten untuk pengguna akhir, tapi juga berintegrasi dengan basis data perusahaan dan aplikasi bisnis.

Aplikasi web menjadi populer karena kemudahan tersedianya aplikasi klien untuk mengaksesnya, penjelajah web, yang kadang disebut sebagai suatu *thin client* (klien tipis). Kemampuan untuk memperbarui dan memelihara aplikasi web tanpa harus mendistribusikan dan menginstalasi perangkat lunak pada kemungkinan ribuan komputer klien merupakan alasan kunci popularitasnya. Aplikasi web yang umum misalnya webmail, toko ritel, lelang, wiki, papan diskusi, weblog, serta MMORPG.

Ketika sebuah organisasi membutuhkan strategi yang penting dari e-commerce, maka dibutuhkan juga aplikasi dengan kualitas yang baik untuk mendapatkan kesuksesan pada pasar e-commerce yang meningkat dengan cepat. Sementara banyak industri dan akademis tertarik pada B2B e-commerce, beberapa definisi tentang apa yang masuk dan tidak pada e-commerce menjadi kontroversial.

(Zwass, 1996) menggunakan definisi yang luas, e-commerce berisi pemakaian bersama informasi bisnis, pemeliharaan hubungan bisnis dan transaksi bisnis dalam jaringan telekomunikasi. (Turban, 2004) mendefinisikan e-commerce sebagai proses pembelian, penjualan atau pertukaran produk, layanan dan informasi melalui jaringan komputer. B2B merupakan sebuah model e-commerce yang semua anggotanya adalah organisasi.

B2B e-commerce mengubah cara sebuah perusahaan menjalani bisnis. Pembangunan aplikasi e-commerce yang efektif dapat meningkatkan komunikasi B2B secara umum dan berpotensi menghasilkan pengurangan biaya dalam jumlah besar. Sangat penting bagi bisnis untuk menggunakan aplikasi yang sesuai untuk membangun transaksi B2Bnya.

Aplikasi web memiliki atribut khusus yang berbeda dari umumnya sistem perangkat lunak. Level medium dari e-commerce adalah world wide web dan transaksi e-commerce merupakan aktivitas terbanyak dari sebuah web. Ketika organisasi berinteraksi melalui internet, maka kualitas e-commerce berhubungan

dengan kualitas web page dan web service. Semua sistem e-commerce berisi banyak dokumen web. Untuk membangun sistem e-commerce yang lebih efektif, harus diketahui faktor desain kritis yang mempengaruhi unjuk kerja aplikasi web. Sehingga kualitas dari aplikasi e-commerce seharusnya mengikuti prinsip yang sama dengan kualitas aplikasi web. Untuk mengidentifikasi atribut kualitas dari aplikasi B2B, maka karakteristik kualitas dari aplikasi web diurutkan pada Tabel 2.3.

Tabel 2.3 Urutan Faktor Kualitas dari Aplikasi Web

<b>Faktor Kualitas</b>	<b>Ranking</b>
Efficiency	1
Security	2
Usability	3
Traceability	4
Availability	5
Scalability	6
Functionality	7
Customizability	8
Recoverability	9
Consistency (Data)	10

Perusahaan yang menghasilkan perangkat lunak telah menggunakan bermacam solusi B2B yang berisi komponen dan karakteristik yang berbeda. Solusi B2B dari enam vendor yang terkenal termasuk Sieble, Oracle, People Soft, IBM, Microsoft, dan SAP telah diteliti oleh banyak peneliti. Karakteristik kualitas dari sistem tersebut diekstrak dan diurutkan berdasarkan frekuensi pengaksesannya pada Tabel 2.4.

Tabel 2.4 Faktor Kualitas dari Aplikasi B2B

<b>Faktor Kualitas</b>	<b>Total Poin</b>	<b>Faktor Kualitas</b>	<b>Total Poin</b>
Security	5	Intractability	2
Scalability	5	Functionality	2
Efficiency	5	Customizability	2
Accessibility	5	Supportability	1
Traceability	3	Open Source Platform	1
Reliability	3	Portability	1
Integrity	3	Generality	1
Usability	2	Changeability	1
Manageability	2	Compatibility	1

Karakteristik kualitas yang penting untuk semua aplikasi web adalah traceability dari transaksi. Juga dibutuhkan availability sistem untuk akses user selama 7x24 jam.

(Behkamal, 2008) mengidentifikasi 44 kriteria penilaian kualitas pada aplikasi B2B yang ditunjukkan pada Tabel 2.5.

Tabel 2.5 Kriteria Kualitas Aplikasi B2B

Kriteria Kualitas	Definisi
Response time	Waktu untuk menyelesaikan sebuah tugas yang spesifik
Throughput	Jumlah task yang dikerjakan dengan sukses pada periode waktu yang diberikan
Turnaround time	Waktu tunggu setelah mengirimkan instruksi untuk memulai sebuah grup dari task yang berhubungan dan pelengkapanya
I/O devices utilization	Waktu dari penempatan I/O devices dibagi dengan waktu khusus yang didesain untuk penempatan I/O devices
Failure avoidance	Jumlah kesalahan pola yang dihindari pada saat desain dan coding dan dibandingkan dengan jumlah kesalahan pola yang disarankan
Mean recovery time	Waktu rata-rata sistem untuk menyelesaikan recovery dari recovery awal sebagian
Restorability	Kemampuan produk untuk restoring dirinya sendiri setelah kejadian tidak normal atau saat request. Kebutuhan restoration contohnya: database checkpoint, transaction checkpoint, redo function, undo function
Memory utilization	Ukuran memori dari produk yang akan ditempatkan untuk menyelesaikan sebuah task yang spesifik
Incorrect operation avoidance	Jumlah fungsi yang diimplementasikan dengan kemampuan menghindari kesalahan operasi
Restartability	Frekuensi sistem dapat melakukan restart untuk menyediakan layanan bagi user pada waktu yang dibutuhkan
Modularity	Kemudahan melakukan update dan generalisasi pengetahuan fungsional berdasarkan fungsi atau data program, sequence dari eksekusi dan hirarki dari aliran kontrol
Computational accuracy	Jumlah perhitungan matematika yang telah diimplementasikan dengan benar dan telah dibuktikan dengan evaluasi kemudian dibandingkan dengan jumlah perhitungan matematika yang dideskripsikan pada spesifikasi kebutuhan
Access auditability	Jumlah tipe akses yang di-log secara benar sebagai n spesifikasi dan dibandingkan dengan jumlah tipe akses yang dibutuhkan untuk disimpan sebagaimana yang dideskripsikan pada spesifikasi
Access controllability	Jumlah kebutuhan access controllability yang diimplementasikan secara benar seperti pada spesifikasi dan dibandingkan dengan jumlah kebutuhan access controllability pada spesifikasi
Completeness of description	Jumlah fungsi yang dideskripsikan secara benar dan dibandingkan dengan jumlah fungsi keseluruhan pada produk
Modification complexity	Kompleksitas fungsi berdasarkan skala yang ditentukan
Globalization	Jumlah implementasi fungsi yang dapat memiliki kemampuan untuk menerima hasil yang dibutuhkan pada berbagai negara yang khusus dan dibandingkan dengan jumlah fungsi dengan kebutuhan kemampuan globalisasi
Ease of installation	Jumlah operasi setup yang implementasinya kompleks dan dibandingkan dengan jumlah operasi setup yang spesifik

Demonstration accessibility	Jumlah demonstrasi/tutorial yang dapat diakses user dengan sukses dibagi dengan jumlah keseluruhan demonstrasi/tutorial yang tersedia
Effectiveness of the user documentation	Jumlah task yang diselesaikan dengan sukses setelah mengakses user documentation dibagi dengan jumlah keseluruhan task yang diuji
Effectiveness of help system in use	Jumlah fungsi yang dapat digunakan dengan benar setelah menggunakan help system
Audit Ability	Kualitas dari produk atau sistem agar setiap orang mampu menganalisa audit secara efektif dan efisien
Audit Trail Capability	Kemampuan untuk mendata semua audit
Clarity	Menampilkan data secara jelas
Communicational Commonality	Penggunaan standarisasi untuk protocol dan interfaces
Completeness	Tingkat kelengkapan data yang diperoleh serta spesifikasi item
Hardware Independence	Tingkat ketergantungan produk pada hardware
Instrumentation	Tingkat pengukuran yang disediakan software yang digunakan untuk mengidentifikasi kesalahan
Complexity	Tingkat kompleksitas dari software
Conciseness	Berhubungan dengan kejelasan dan kesingkatan source code dalam bentuk lines of code
Correctness	Kebenaran dari spesifikasi suatu fungsi
Data Commonality	Penggunaan representasi standar data
Degradability	Tingkat penurunan kemampuan dari produk
Expandability	Tingkat kebutuhan penyimpanan dari fungsi software yang dapat dikembangkan
Hardware Environmental Adaptability	Kemampuan produk untuk beradaptasi dengan perubahan lingkungan hardware yang berhubungan
Organizational Environment Adaptability	Kemampuan produk untuk beradaptasi pada perubahan organisasi
Reusability	Kemudahan menggunakan kembali bagian tertentu dengan sedikit perubahan
Robustness	Tingkat ketahanan perangkat lunak terhadap kejadian tak terduga
Self-Description Documentation	Merupakan kumpulan deskripsi yang menerangkan mengenai kemampuan dari komponen
Simplicity	Tingkat kesederhanaan produk
Software Independence	Tingkat ketidaktergantungan software dengan lingkungannya, konstruksi non standart language, sistem operasi, libraries, sistem manajemen basis data,dll
Trust Ability	Kemampuan produk dipercaya
Uniformity	Penggunaan standar pada produk
Withdraw Ability	Kemampuan pemindahan property dari element

Kemudian dilakukan pemetaan antara kriteria kualitas aplikasi B2B dengan faktor kualitas dari ISO 9126. Tabel 2.6 berisi hubungan antara faktor kualitas dari ISO 9126 dengan kriteria kualitas untuk mengevaluasi aplikasi B2B.

Tabel 2.6 Faktor Kualitas vs Kriteria Kualitas

Faktor Kualitas	Kriteria	Faktor Kualitas	Kriteria
Functionality	Access audit ability	Maintainability	Audit ability
	Access controllability		Audit trail capability
	Audit ability		Completeness of description
	Clarity		Complexity
	Communication Commonality		Modification complexity
	Completeness		Modularity
	Computational Accuracy		Reusability
	Correctness		Self-Description Documentation
	Data Commonality	Portability	Ease of installation
Reliability	Uniformity		Expendability
	Degradability		Hardware environmental adaptability
	Failure avoidance		Hardware Independence
	Incorrect operation avoidance		Organizational environment adaptability
	Instrumentation		Software Independence
	Mean recovery time		Withdraw ability
	Restart ability	Usability	Access audit ability
	Restorability		Clarity
	Robustness		Completeness of description
Efficiency	Trust ability		Demonstration accessibility
	I/O devices Utilization		Effectiveness of help System in use
	Memory utilization		Effectiveness of the user documentation
	Response time		Globalization
	Throughput		Simplicity
	Turnaround Time		Uniformity

Bobot digunakan sebagai dasar penghitungan hasil akhir kualitas aplikasi web, sehingga nilai bobot akan berpengaruh pada prosentase kualitas aplikasi web. Metode yang digunakan untuk pembobotan pada penelitian ini adalah FAHP.

## 2.5 Fuzzy Analytical Hierarchycal Process (FAHP)

Bilangan fuzzy adalah sebuah fuzzy subset dari bilangan real, menyatakan pengembangan konsep rentang kepercayaan. Sebuah triangular fuzzy number (TFN) memiliki ciri-ciri dasar seperti dibawah ini.

Sebuah bilangan fuzzy  $\tilde{A}$  pada 3 adalah TFN bila fungsi keanggotaannya  $\mu_{\tilde{A}}(x) : 3 \rightarrow [0,1]$  adalah sama dengan:

$$\mu_{\tilde{A}}(x) = \begin{cases} \frac{x-L}{M-L}, & L \leq x \leq M \\ \frac{U-x}{U-M}, & M \leq x \leq U \\ 0, & \text{lainnya} \end{cases} \quad (2.1)$$



Dimana L dan U adalah batas bawah dan batas atas bilangan fuzzy  $\tilde{A}$ , sedangkan M adalah nilai tengah. TFN dapat dinotasikan dengan  $\tilde{A}=(L,M,U)$ , dan berikut adalah hukum operasi pada dua TFN  $\tilde{A}_1 = (L_1, M_1, U_1)$  dan  $\tilde{A}_2 = (L_2, M_2, U_2)$ .

a. Penjumlahan

$$\begin{aligned}\tilde{A}_1 \oplus \tilde{A}_2 &= (L_1, M_1, U_1) \oplus (L_2, M_2, U_2) \\ &= (L_1 + L_2, M_1 + M_2, U_1 + U_2)\end{aligned}\quad (2.2.)$$

b. Perkalian

$$\begin{aligned}\tilde{A}_1 \otimes \tilde{A}_2 &= (L_1, M_1, U_1) \otimes (L_2, M_2, U_2) \\ &= (L_1 L_2, M_1 M_2, U_1 U_2)\end{aligned}\quad (2.3)$$

c. Pengurangan

$$\begin{aligned}\tilde{A}_1 - \tilde{A}_2 &= (L_1, M_1, U_1) - (L_2, M_2, U_2) \\ &= (L_1 - U_2, M_1 - M_2, U_1 + L_2)\end{aligned}\quad (2.4)$$

d. Pembagian

$$\begin{aligned}\tilde{A}_1 / \tilde{A}_2 &= (L_1, M_1, U_1) / (L_2, M_2, U_2) \\ &= (L_1 / U_2, M_1 / M_2, U_1 + L_2) \\ &\text{untuk } L_i > 0, M_i > 0, U_i > 0\end{aligned}\quad (2.5)$$

e. Inversi

$$\begin{aligned}\tilde{A}_1^{-1} &= (L_1, M_1, U_1)^{-1} = (1 / L_1, 1 / M_1, 1 / U_1) \\ &\text{untuk } L_i > 0, M_i > 0, U_i > 0\end{aligned}\quad (2.6)$$

Variabel linguistik adalah sebuah variabel dimana nilainya berupa kata-kata atau kalimat dalam bahasa alami atau buatan. Teknik komputasinya didasarkan pada bilangan fuzzy yang didefinisikan seperti Tabel 2.7. Setiap fungsi keanggotaan (skala bilangan fuzzy) didefinisikan oleh tiga parameter TFN simetris, titik kiri, titik tengah dan titik kanan pada interval dimana fungsi tersebut didefinisikan. Penggunaan variabel linguistik disini ditujukan untuk mengkaji prioritas linguistik yang diberikan oleh evaluator.

Tabel 2.7 Fungsi Keanggotaan Skala Linguistik

Bilangan Fuzzy	Skala Linguistik	Skala Bilangan Fuzzy
$\tilde{1}$	Sama Penting	(1,1,3)
$\tilde{3}$	Sedikit Lebih Penting	(1,3,5)
$\tilde{5}$	Lebih Penting	(3,5,7)
$\tilde{7}$	Sangat Penting	(5,7,9)
$\tilde{9}$	Paling Penting	(7,9,9)

Jika perbandingan dua kriteria tidak memenuhi skala diatas tetapi merupakan kebalikannya, maka digunakan inversi seperti pada Tabel 2.8.

Tabel 2.8 Fungsi Keanggotaan Skala Linguistik Inversi

Bilangan Fuzzy	Skala Linguistik	Skala Bilangan Fuzzy
$\tilde{1}^{-1}$	Sama Tidak Penting	(1/3,1,1)
$\tilde{3}^{-1}$	Sedikit Lebih Tidak Penting	(1/5,1/3,1)
$\tilde{5}^{-1}$	Lebih Tidak Penting	(1/7,1/5,1/3)
$\tilde{7}^{-1}$	Sangat Tidak Penting	(1/9,1/7,1/5)
$\tilde{9}^{-1}$	Paling Tidak Penting	(1/9,1/9,1/7)

Prosedur untuk menjelaskan bobot kriteria evaluasi dengan FAHP dapat diterangkan sebagai berikut:

1. Menyusun matrik perbandingan berpasangan diantara semua elemen/kriteria dalam dimensi sistem hirarki berdasarkan penilaian dengan variabel linguistik.

$$\tilde{A} = \begin{bmatrix} 1 & \tilde{a}_{12} & \dots & \tilde{a}_{1n} \\ \tilde{a}_{21} & 1 & \dots & \tilde{a}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{a}_{n1} & \tilde{a}_{n2} & \dots & 1 \end{bmatrix} \quad (2.7)$$

Dimana:

$$\tilde{a}_{ij} = \begin{cases} \tilde{1}, \tilde{3}, \tilde{5}, \tilde{7}, \tilde{9} & \text{kriteria i relatif penting terhadap j} \\ 1 & i=j \\ \tilde{1}^{-1}, \tilde{3}^{-1}, \tilde{5}^{-1}, \tilde{7}^{-1}, \tilde{9}^{-1} & \text{kriteria i relatif kurang penting terhadap j} \end{cases}$$

2. Mendefinisikan rata-rata geometris fuzzy dan bobot fuzzy setiap kriteria dengan rata-rata.

$$\tilde{r}_i = \tilde{a}_{i1} \otimes \tilde{a}_{i2} \otimes \dots \otimes \tilde{a}_{in} \quad (2.8)$$

$$\tilde{w}_i = \tilde{r}_i \otimes (\tilde{r}_1 \oplus \dots \oplus \tilde{r}_n)^{-1} \quad (2.9)$$

Dimana  $\tilde{a}_{in}$  adalah nilai perbandingan fuzzy dari kriteria i ke n,  $\tilde{r}_i$  adalah rata-rata geometris dari nilai perbandingan fuzzy kriteria i terhadap setiap kriteria dan  $\tilde{w}_i$  adalah bobot fuzzy dari kriteria ke i dan dapat diindikasikan dengan TFN  $\tilde{w}_i = (Lw_i, Mw_i, Uw_i)$ .  $Lw_i$ ,  $Mw_i$  dan  $Uw_i$  masing-masing adalah nilai bawah, tengah dan atas dari bobot fuzzy kriteria ke i.

3. Melakukan de-fuzzy-fikasi dari hasil bobot fuzzy pada langkah ke-2 dengan menggunakan persamaan (2.10).

$$Lw_i + ((Mw_i - Lw_i) + (Uw_i - Lw_i)) / 3 \quad (2.10)$$

## **BAB 3**

### **METODA PENELITIAN**

Untuk membangun CASE tool untuk penilaian kualitas berdasarkan ISO 9126 dengan memanfaatkan karakteristik dari aplikasi web maka perlu dilakukan tahapan seperti terlihat pada Gambar 3.1.



Gambar 3.1 Blok Diagram Pengerjaan Penelitian

#### **3.1 Pembuatan Panduan Penilaian Kualitas**

Dibuatkan panduan bagi pengguna agar penilaian kualitas aplikasi B2B menjadi lebih mudah dan terukur. Panduan berisi komputasi, interpretasi, skala dan skala penilaian seperti pada Tabel 3.1. Komputasi berupa perhitungan dengan melibatkan variabel yang berpengaruh pada nilai kualitas. Interpretasi adalah penilaian terhadap hasil komputasi yang kemudian dibagi dalam range penilaian 0 sampai 3 pada skala penilaian. 0 adalah nilai minimum dan 3 merupakan nilai maksimum dari nilai kualitas. Skala penilaian ini disusun berdasarkan interpretasi dan komputasi dengan mempertimbangkan definisi dari kriteria kualitas. Pada skala

terdapat nilai absolut yang berarti nilai mutlak, ratio berarti perbandingan dan ordinal berarti urutan (order).

Dari 44 jumlah kriteria kualitas, tidak semuanya berdasarkan referensi. Kriteria kualitas dengan tanda \*) dan \*\*) dibuat dan dilengkapi berdasarkan definisinya serta kesamaan dengan kriteria yang lain.

Tabel 3.1 Panduan Penilaian Kriteria

Kriteria Kualitas	Komputasi	Interpretasi	Status	Skala Penilaian
Audit ability	$X=A/B$ A:Jumlah data yang dapat diaudit B:Jumlah data keseluruhan	$0 \leq X \leq 1$ , makin mendekati 1 makin mudah diaudit	A	0: $0 \leq x \leq 0,25$ ; 1: $0,25 < x \leq 0,5$ ; 2: $0,5 < x \leq 0,75$ ; 3: $0,75 < x \leq 1$ ;
Access auditability	$X=A/B$ A:Jumlah tipe akses yang telah disimpan B:Jumlah tipe akses yang dibutuhkan untuk disimpan	$0 \leq X \leq 1$ , makin mendekati 1 makin mudah diaudit	A	0: $x=0-0,25$ ; 1: $x=0,26-0,5$ ; 2: $x=0,51-0,75$ ; 3: $x=0,76-1$
Access controllability	$X=A/B$ A:Jumlah kebutuhan access controllability yang diimplementasikan secara benar B:Jumlah kebutuhan access controllability pada spesifikasi	$0 \leq X \leq 1$ , makin mendekati 1 makin terkontrol	A	0: $x=0-0,25$ ; 1: $x=0,26-0,5$ ; 2: $x=0,51-0,75$ ; 3: $x=0,76-1$
Audit Trail Capability	$X=A/B$ A:Jumlah data audit B:Jumlah data keseluruhan	$0 \leq X \leq 1$ , makin mendekati 1 makin lengkap data audit	A	0: $0 \leq x \leq 0,25$ ; 1: $0,25 < x \leq 0,5$ ; 2: $0,5 < x \leq 0,75$ ; 3: $0,75 < x \leq 1$ ;
Clarity	$X=A/B$ A: jumlah fungsi yang diimplementasikan dengan penjelasannya B: jumlah fungsi keseluruhan	$0 \leq X \leq 1$ makin mendekati 1 makin jelas	A	0: $0 \leq x \leq 0,25$ ; 1: $0,25 < x \leq 0,5$ ; 2: $0,5 < x \leq 0,75$ ; 3: $0,75 < x \leq 1$ ;
Communicational Commonality	$X=A/B$ A: jumlah bagian yang sesuai standar B: jumlah produk keseluruhan	$0 \leq X \leq 1$ , makin besar dan mendekati 1 lebih baik, makin banyak distandarisasi	A	0: $0 \leq x \leq 0,25$ ; 1: $0,25 < x \leq 0,5$ ; 2: $0,5 < x \leq 0,75$ ; 3: $0,75 < x \leq 1$ ;
Completeness	$X=1-A/B$ A: jumlah fungsi hilang yang terdeteksi selama evaluasi B: jumlah fungsi yang dideskripsikan saat spesifikasi kebutuhan	$0 \leq X \leq 1$ makin mendekati 1 makin lengkap	A	0: $0 \leq x \leq 0,25$ ; 1: $0,25 < x \leq 0,5$ ; 2: $0,5 < x \leq 0,75$ ; 3: $0,75 < x \leq 1$ ;
Completeness of description	$X=A/B$ A:Jumlah fungsi atau tipe fungsi yang dijelaskan pada deskripsi produk B:Jumlah fungsi atau tipe	$0 \leq X \leq 1$ , makin mendekati 1 makin lengkap	A	0: $x=0-0,25$ ; 1: $x=0,26-0,5$ ; 2: $x=0,51-0,75$ ; 3: $x=0,76-1$

	fungsi			
Complexity	$X = \text{Complexity}(\text{function})$	$0 < X$ , makin menjauhi ordinal makin baik	NA	0: $x < 1$ ; 1: $1 \leq x \leq 3$ ; 2: $3 < x \leq 5$ ; 3: $x > 5$
Computational accuracy	$X = A/B$ A:Jumlah perhitungan matematika yang akurat B:Jumlah perhitungan matematika yang dideskripsikan pada spesifikasi kebutuhan	$0 \leq X \leq 1$ , makin mendekati 1 makin akurat	A	0: $x = 0-0,25$ ; 1: $x = 0,26-0,5$ ; 2: $x = 0,51-0,75$ ; 3: $x = 0,76-1$
Conciseness	$X = \text{Complexity}(\text{function})$	$0 < X$ , makin menjauhi ordinal makin baik	NA	0: $x < 1$ ; 1: $1 \leq x \leq 3$ ; 2: $3 < x \leq 5$ ; 3: $x > 5$
Correctness *)	$X = 1 - A/B$ A: jumlah fungsi yang hilang atau salah implementasi B: jumlah fungsi yang dideskripsikan saat spesifikasi kebutuhan	$0 \leq X \leq 1$ makin mendekati 1 makin benar	A	0: $0 \leq x \leq 0,25$ ; 1: $0,25 < x \leq 0,5$ ; 2: $0,5 < x \leq 0,75$ ; 3: $0,75 < x \leq 1$ ;
Data Commonality )	$X = A/B$ A: jumlah data yang sesuai standar B: jumlah data keseluruhan	$0 \leq X \leq 1$ makin mendekati 1 makin baik	A	0: $0 \leq x \leq 0,25$ ; 1: $0,25 < x \leq 0,5$ ; 2: $0,5 < x \leq 0,75$ ; 3: $0,75 < x \leq 1$ ;
Degradability *)	$X = A/B$ A: tingkat penurunan B: umur dari produk	$0 \leq X \leq 1$ , makin kecil dan menjauhi 1 makin baik	A	0: $0,75 < x \leq 1$ ; 1: $0,5 < x \leq 0,75$ ; 2: $0,25 < x \leq 0,5$ - 0,5; 3: $0 \leq x \leq 0,25$ ;
Demonstration accessibility	$X = A/B$ A:Jumlah demonstrasi/tutorial yang dapat diakses user dengan sukses B:jumlah keseluruhan demonstrasi/tutorial yang tersedia	$0 \leq X \leq 1$ , makin mendekati 1 makin baik	A	0: $x = 0-0,25$ ; 1: $x = 0,26-0,5$ ; 2: $x = 0,51-0,75$ ; 3: $x = 0,76-1$
Ease of installation	$X = A/B$ A:Jumlah fungsi yang memiliki operasi setup yang implementasinya kompleks dan dibuktikan dengan evaluasi B:Jumlah operasi setup khusus secara keseluruhan	$0 \leq X \leq 1$ , makin mendekati 1 makin mudah	A	0: $x = 0-0,25$ ; 1: $x = 0,26-0,5$ ; 2: $x = 0,51-0,75$ ; 3: $x = 0,76-1$
Effectiveness of help system in use	$X = A/B$ A:Jumlah fungsi yang dapat digunakan B:Jumlah keseluruhan fungsi yang disediakan	$0 \leq X \leq 1$ , makin mendekati 1 makin baik	A	0: $x = 0-0,25$ ; 1: $x = 0,26-0,5$ ; 2: $x = 0,51-0,75$ ; 3: $x = 0,76-1$
Effectiveness of the user documentation	$X = A/B$ A:Jumlah task yang diselesaikan dengan sukses B:Jumlah keseluruhan task yang diuji	$0 \leq X \leq 1$ , makin mendekati 1 makin baik	A	0: $x = 0-0,25$ ; 1: $x = 0,26-0,5$ ; 2: $x = 0,51-0,75$ ; 3: $x = 0,76-1$
Expandability *)	$X = A/B$	$0 \leq X \leq 1$	NA	0: $0 \leq x \leq 0,25$ ;

	A: jumlah kebutuhan yang dapat dipenuhi B: jumlah kebutuhan keseluruhan	makin mendekati 1 makin benar		1: $0,25 < x \leq 0,5$ ; 2: $0,5 < x \leq 0,75$ ; 3: $0,75 < x \leq 1$ ;
Failure avoidance	$X=A/B$ A:Jumlah kesalahan pola yang dapat dihindari B:Jumlah kesalahan pola yang disarankan	$0 \leq X \leq 1$ , makin mendekati 1 makin baik, user dapat lebih sering menghindari kesalahan pola	A	0: $x=0-0,25$ ; 1: $x=0,26-0,5$ ; 2: $x=0,51-0,75$ ; 3: $x=0,76-1$
Globalization *)	$X=A/B$ A:Jumlah fungsi yang mampu memperoleh globalisasi dan dibuktikan dengan evauasi B:Jumlah fungsi keseluruhan	$0 \leq X \leq 1$ , makin mendekati 1 makin baik	A	0: $x=0-0,25$ ; 1: $x=0,26-0,5$ ; 2: $x=0,51-0,75$ ; 3: $x=0,76-1$
Hardware Environmental Adaptability	$X=A/B$ A: jumlah implementasi fungsi yang mampu menerima hasil yang dibutuhkan pada lingkungan hardware yang beragam B: jumlah fungsi dengan kebutuhan kemampuan adaptasi lingkungan hardware	$0 \leq X \leq 1$ makin mendekati 1 makin baik	A	0: $0 \leq x \leq 0,25$ ; 1: $0,25 < x \leq 0,5$ ; 2: $0,5 < x \leq 0,75$ ; 3: $0,75 < x \leq 1$ ;
Hardware Independence	$X=A/B$ A: jumlah produk yang tergantung pada hardware B: jumlah produk keseluruhan	$0 \leq X \leq 1$ , makin kecil dan menjauhi 1 makin baik	A	0: $0,75 < x \leq 1$ ; 1: $0,5 < x \leq 0,75$ ; 2: $0,25 < x \leq 0,5$ - 0,5; 3: $0 \leq x \leq 0,25$ ;
I/O devices utilization	$X=A/B$ A:Waktu dari penempatan I/O devices B:waktu khusus yang didesain untuk penempatan I/O devices	$0 \leq X \leq 1$ , makin kecil dan menjauhi 1 makin baik	A	0: $x=0,76-1$ ; 1: $x=0,51-0,75$ ; 2: $x=0,26-0,5$ ; 3: $x=0-0,25$ ;
Incorrect operation avoidance	$X=A/B$ A:Jumlah kesalahan kritis dan serius B:Jumlah kasus percobaan yang dijalankan dari pola kesalahan operasi selama percobaan (kebanyakan menyebabkan kesalahan)	$0 \leq X \leq 1$ , makin mendekati 1 makin baik, lebih banyak operasi kesalahan user yang dihindari	A	0: $x=0-0,25$ ; 1: $x=0,26-0,5$ ; 2: $x=0,51-0,75$ ; 3: $x=0,76-1$
Instrumentation	$X=A/B$ A: jumlah fungsi yang dapat diukur kesalahannya B: jumlah fungsi keseluruhan	$0 \leq X \leq 1$ makin mendekati 1 makin jelas	A	0: $0 \leq x \leq 0,25$ ; 1: $0,25 < x \leq 0,5$ ; 2: $0,5 < x \leq 0,75$ ; 3: $0,75 < x \leq 1$ ;
Mean recovery time	$X=Sum(T)/B$ T:Waktu untuk recovery sistem software yang down pada tiap kesempatan B:Jumlah kasus yang diobservasi sistem software telah masuk pada recovery	$0 \leq X$ , makin kecil makin baik	A	0: $t > 3,1$ ; 1: $t=2-3$ ; 2: $t=1-1,9$ ; 3: $t=0-0,9$
Memory utilization **)	Ukuran dalam byte	$0 \leq X$ makin kecil makin	NA	0: $0,75 < x \leq 1$ ; 1: $0,5 < x \leq 0,75$ ; 2: $0,25 < x \leq 0,5$ -

		baik		0,5; 3: $0 \leq x \leq 0,25$ ;
Modification complexity	$X = \text{Complexity}(\text{function})$	$0 < X$ , makin menjauhi ordinal makin baik	NA	0: $x < 1$ ; 1: $x = 1-3$ ; 2: $x = 3,1-5$ ; 3: $x > 5$
Modularity **)	$X1 = A1/B1$ $X2 = A2/B2$ A1:Jumlah modul yang berhubungan satu sama lain secara fungsional A2:Jumlah modul yang berhubungan satu sama lain pada struktur data B1,B2:jumlah modul	$0 \leq X \leq 1$ , makin besar dan mendekati 1 lebih baik	A	0: $0 \leq x \leq 0,25$ ; 1: $0,25 < x \leq 0,5$ ; 2: $0,5 < x \leq 0,75$ ; 3: $0,75 < x \leq 1$ ;
Organizational Environment Adaptability	$X = A/B$ A: jumlah implementasi fungsi yang mampu menerima hasil yang dibutuhkan pada lingkungan organisasi dan bisnis yang beragam B: jumlah fungsi dengan kebutuhan kemampuan adaptasi lingkungan organisasi	$0 \leq X \leq 1$ makin mendekati 1 makin baik	A	0: $0 \leq x \leq 0,25$ ; 1: $0,25 < x \leq 0,5$ ; 2: $0,5 < x \leq 0,75$ ; 3: $0,75 < x \leq 1$ ;
Response time	$T = A - B$ A:Waktu untuk memperoleh hasil B:Waktu setelah perintah selesai dimasukkan	$0 < T$ , makin cepat makin baik	A	0: $t = 0,1-0,9$ ; 1: $t = 1-1,9$ ; 2: $t > 3,1$ ; 3: $t = 2-3$
Restartability	$X = A/B$ A:Jumlah restart yang sesuai dengan waktu yang dibutuhkan selama testing atau operasi user pendukung B:Jumlah restart keseluruhan selama testing atau operasi user pendukung	$0 \leq X \leq 1$ , makin besar dan mendekati 1 makin baik, user dapat melakukan restart dengan mudah	A	0: $x = 0-0,25$ ; 1: $x = 0,26-0,5$ ; 2: $x = 0,51-0,75$ ; 3: $x = 0,76-1$
Restorability	$X = A/B$ A:Jumlah kasus restore yang sukses dilakukan B:Jumlah kasus restore yang diuji tiap requirement	$0 \leq X \leq 1$ , makin besar dan mendekati 1 lebih baik, produk lebih mampu untuk restore pada kasus yang didefinisikan	A	0: $x = 0-0,25$ ; 1: $x = 0,26-0,5$ ; 2: $x = 0,51-0,75$ ; 3: $x = 0,76-1$
Reusability *)	$X = A/B$ A:Jumlah bagian yang dapat digunakan kembali B:Jumlah produk keseluruhan	$0 \leq X \leq 1$ , makin besar dan mendekati 1 lebih baik, makin banyak yang dapat digunakan ulang	A	0: $0 \leq x \leq 0,25$ ; 1: $0,25 < x \leq 0,5$ ; 2: $0,5 < x \leq 0,75$ ; 3: $0,75 < x \leq 1$ ;
Robustness *)	$X = A/B$ A:Jumlah bagian yang memiliki ketahanan B:Jumlah produk keseluruhan	$0 \leq X \leq 1$ , makin besar dan mendekati 1 lebih baik	A	0: $0 \leq x \leq 0,25$ ; 1: $0,25 < x \leq 0,5$ ; 2: $0,5 < x \leq 0,75$ ; 3: $0,75 < x \leq 1$ ;
Self-Description	$X = A/B$ A:Jumlah deskripsi	$0 \leq X \leq 1$ , makin mendekati 1	A	0: $0 \leq x \leq 0,25$ ; 1: $0,25 < x \leq 0,5$ ;



Documentation *)	B:Jumlah semua komponen	makin lengkap		2: $0,5 < x \leq 0,75$ ; 3: $0,75 < x \leq 1$ ;
Simplicity *)	$X=A/B$ A:Jumlah bagian yang sederhana B:Jumlah produk keseluruhan	$0 \leq X \leq 1$ , makin besar dan mendekati 1 lebih baik	NA	0: $0 \leq x \leq 0,25$ ; 1: $0,25 < x \leq 0,5$ ; 2: $0,5 < x \leq 0,75$ ; 3: $0,75 < x \leq 1$ ;
Software Independence *)	$X=A/B$ A: jumlah produk yang tergantung pada software B: jumlah produk keseluruhan	$0 \leq X \leq 1$ , makin kecil dan menjauhi 1 makin baik	A	0: $0,75 < x \leq 1$ ; 1: $0,5 < x \leq 0,75$ ; 2: $0,25 < x \leq 0,5$ ; 0,5; 3: $0 \leq x \leq 0,25$ ;
Throughput	$X=A/B$ A:Jumlah task yang lengkap B:Periode waktu observasi	$0 < X$ , lebih besar lebih baik	A	0: $x < 1$ ; 1: $x = 1-3$ ; 2: $x = 3,1-5$ ; 3: $x > 5$
Trust Ability *)	$X=A/B$ A: jumlah bagian yang terpercaya B: jumlah produk keseluruhan	$0 \leq X \leq 1$ , makin besar dan mendekati 1 lebih baik	NA	0: $0 \leq x \leq 0,25$ ; 1: $0,25 < x \leq 0,5$ ; 2: $0,5 < x \leq 0,75$ ; 3: $0,75 < x \leq 1$ ;
Turnaround time	$X=A-B$ A:Waktu untuk memperoleh output akhir B:Waktu penyelesaian request	$0 < T$ , lebih pendek lebih baik	A	0: $t = 0,1-0,9$ ; 1: $t = 1-1,9$ ; 2: $t > 3,1$ ; 3: $t = 2-3$
Uniformity *)	$X=A/B$ A: jumlah bagian yang sesuai standar B: jumlah produk keseluruhan	$0 \leq X \leq 1$ , makin besar dan mendekati 1 lebih baik	A	0: $0 \leq x \leq 0,25$ ; 1: $0,25 < x \leq 0,5$ ; 2: $0,5 < x \leq 0,75$ ; 3: $0,75 < x \leq 1$ ;
Withdraw Ability *)	$X=A/B$ A: jumlah bagian yang dapat dipindahkan B: jumlah produk keseluruhan	$0 \leq X \leq 1$ , makin besar dan mendekati 1 lebih baik, makin tinggi kemampuan pemindahannya	NA	0: $0 \leq x \leq 0,25$ ; 1: $0,25 < x \leq 0,5$ ; 2: $0,5 < x \leq 0,75$ ; 3: $0,75 < x \leq 1$ ;

Keterangan:

\*) Data tidak ada

\*\*) Data ada tapi belum lengkap

A: Applicable

NA: Not Applicable

### 3.2 Pembuatan Saran untuk Perbaikan Kualitas

Dari hasil penilaian kualitas akan diperoleh nilai kualitas dari sebuah sistem aplikasi. CASE Tool ini dirancang untuk dapat memberikan saran bagi perbaikan kualitas seperti pada Tabel 3.2. Saran dibuat untuk tiap-tiap kriteria kualitas berdasarkan definisinya yang telah dijelaskan pada Tabel 2.4

Tabel 3.2 Saran untuk Perbaikan Kualitas

Kriteria Kualitas	Saran Perbaikan
Response time	Percepat waktu penyelesaian tugas yang spesifik
Throughput	Tingkatkan jumlah task yang dapat dikerjakan dengan sukses pada periode tertentu
Turnaround time	Perpendek waktu tunggu setelah pengiriman instruksi untuk memulai sebuah grup dari task yang berhubungan dan pelengkapanya
I/O devices utilization	Perpendek waktu dari penempatan I/O devices
Failure avoidance	Tingkatkan jumlah kesalahan pola yang dapat dihindari pada saat desain dan coding
Mean recovery time	Perpendek waktu rata-rata sistem untuk menyelesaikan recovery
Restorability	Tingkatkan kemampuan produk untuk restoring dirinya sendiri setelah kejadian tidak normal atau saat request.
Memory utilization	Perkecil ukuran memori dari produk yang akan ditempatkan untuk menyelesaikan sebuah task yang spesifik
Incorrect operation avoidance	Tingkatkan jumlah implementasi fungsi yang memiliki kemampuan menghindari kesalahan operasi
Restartability	Tingkatkan kemampuan sistem untuk melakukan restart pada waktu yang dibutuhkan selama testing atau operasi user pendukung
Modularity	Tingkatkan kemudahan melakukan update dan generalisasi pengetahuan fungsional berdasarkan fungsi atau data program, sequence dari eksekusi dan hirarki dari aliran kontrol
Computational accuracy	Tingkatkan jumlah perhitungan matematika yang dapat diimplementasikan dengan benar
Access auditability	Tingkatkan kemampuan untuk melakukan pengaksesan audit
Access controllability	Tingkatkan jumlah kebutuhan access controllability yang dapat diimplementasikan secara benar
Completeness of description	Tingkatkan jumlah fungsi yang dideskripsikan secara benar
Modification complexity	Tingkatkan kompleksitas fungsi berdasarkan skala tertentu
Globalization	Tingkatkan jumlah implementasi fungsi yang dapat memiliki kemampuan untuk menerima hasil yang dibutuhkan pada berbagai negara khusus
Ease of installation	Tingkatkan kemudahan operasi setup yang implementasinya kompleks
Demonstration accesibility	Tingkatkan jumlah demonstrasi/tutorial yang dapat diakses user dengan sukses
Effectiveness of the user documentation	Tingkatkan jumlah task yang diselesaikan dengan sukses setelah mengakses user documentation
Effectiveness of help system in use	Tingkatkan jumlah fungsi yang dapat digunakan dengan benar setelah menggunakan help system
Audit Ability	Tingkatkan kemampuan produk untuk diaudit
Audit Trail Capability	Tingkatkan kemampuan untuk mendata semua audit
Clarity	Tingkatkan kejelasan fungsi
Communicational Commonality	Tingkatkan penggunaan standarisasi untuk protocol dan interfaces
Completeness	Tingkatkan kelengkapan data serta spesifikasi item
Hardware Independence	Kurangi tingkat ketergantungan software pada hardware
Instrumentation	Tingkatkan jumlah pengukuran yang disediakan software untuk mengidentifikasi kesalahan

Complexity	Tingkatkan kompleksitas dari software
Conciseness	Tingkatkan kejelasan dan kesingkatan source code dalam bentuk lines of code
Correctness	Tingkatkan kebenaran dari spesifikasi fungsi
Data Commonality	Tingkatkan Penggunaan representasi standar data
Degradability	Meminimalisasi tingkat penurunan kemampuan dari produk
Expandability	Meningkatkan kemampuan produk untuk mencukupi kebutuhan penyimpanan dari fungsi software yang dapat dikembangkan
Hardware Environmental Adaptability	Tingkatkan kemampuan produk untuk beradaptasi dengan perubahan lingkungan hardware yang berhubungan
Organizational Environment Adaptability	Tingkatkan kemampuan produk untuk beradaptasi pada perubahan organisasi
Reusability	Tingkatkan kemudahan menggunakan kembali bagian tertentu dengan sedikit perubahan
Robustness	Tingkatkan ketahanan perangkat lunak terhadap kejadian tak terduga
Self-Description Documentation	Tingkatkan jumlah kumpulan deskripsi yang menerangkan mengenai kemampuan dari komponen
Simplicity	Tingkatkan kesederhanaan produk
Software Independence	Tingkatkan ketidaktergantungan software dengan lingkungannya
Trust Ability	Tingkatkan kemampuan produk untuk dipercaya
Uniformity	Tingkatkan penggunaan standar pada produk
Withdraw Ability	Tingkatkan kemampuan pemindahan property dari element

### 3.3 Pemetaan Kriteria ke Sub-Faktor Kualitas

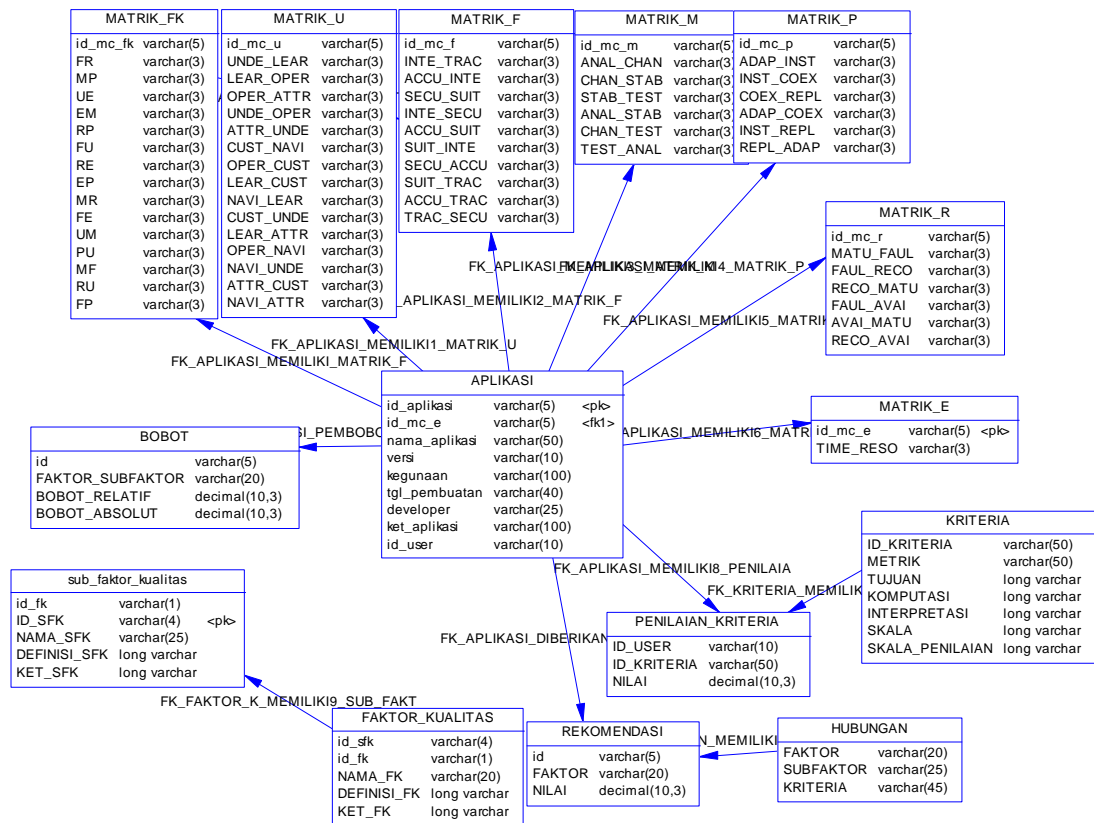
Untuk menghasilkan nilai total kualitas sistem, akan dilakukan perkalian antara bobot absolut dari sub-faktor kualitas dengan nilai kriteria. (Behkamal, 2008) hanya memetakan kriteria ke faktor kualitas. Maka pada penelitian ini, dilakukan pengembangan dengan memetakan kriteria ke sub-faktor kualitas. Hasil pemetaan terdapat pada Tabel 3.3.

Tabel 3.3 Pemetaan Kriteria ke Sub-Faktor Kualitas

Faktor Kualitas	Sub-Faktor Kualitas	Kriteria
Functionality	Suitability	Completeness
	Accuracy	Computational Accuracy
		Correctness
	Interoperability	Communicational Commonality
		Uniformity
	Security	Access Audit Ability
		Access Controllability

	Traceability	Audit Ability
		Clarity
		Data Commonality
Reliability	Fault Tolerance	Failure Avoidance
		Incorrect Operation Avoidance
	Maturity	Robustness
		Trust Ability
		Degradability
	Recoverability	Mean Recovery Time
		Restartability
		Restoreability
	Availability	Instrumentation
Usability	Understandability	Demonstration Accessibility
		Access Audit Ability
	Learnability	Effectiveness of help system in use
		Effectiveness of the user documentation
	Operability	Audit Ability
		Uniformity
	Attractiveness	Simplicity
	Customizability	Globalization
Efficiency	Time Behavior	Response Time
		Throughput
		Turnaround Time
	Resource Utilization	I/O Devices Utilization
		Memory Utilization
Maintainability	Analysability	Audit Trail Capability
		Audit Ability
		Completeness of Description
	Changeability	Modification Complexity
		Complexity
		Self-Description Documentation
	Stability	Reusability
	Testability	Modularity
Portability	Replaceability	Expandability
	Co-Existence	Withdraw Ability
	Installability	Ease of Installation
	Adaptability	Hardware Independence





Gambar 3.3 Physical Data Model

Tabel 3.4 Keterangan Tabel

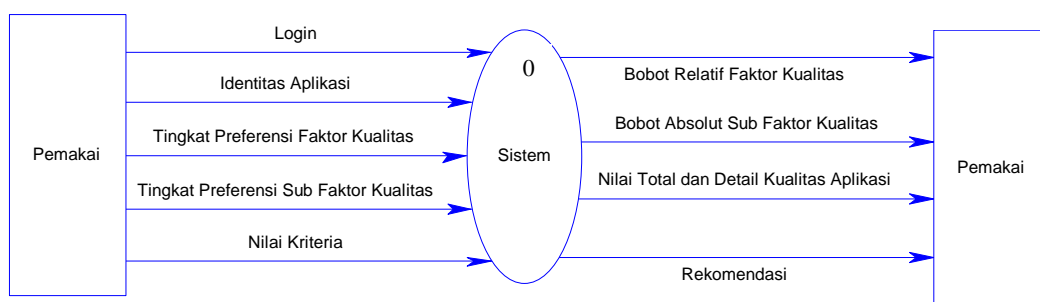
Nama Tabel	Keterangan
Aplikasi	identitas aplikasi yang akan dinilai kualitasnya
faktor_kualitas	nama dan definisi dari 6 faktor kualitas
sub_faktor_kualitas	nama dan definisi dari 25 sub faktor kualitas
matrik_p	masukan pengguna saat membandingkan tingkat preferensi dari empat sub faktor kualitas portability yaitu: adaptability, installability, co-existence dan replaceability. Contohnya pada field ADAP_INST yang berisi masukan tingkat preferensi sub faktor adaptability dengan installability.
matrik_f	tingkat preferensi dari lima sub faktor kualitas functionality yaitu: interoperability, traceability, accuracy, security dan suitability
matrik_e	perbandingan tingkat preferensi sub faktor dari efficiency, yaitu time behavior dan resource utilization
matrik_r	perbandingan tingkat preferensi antara empat sub faktor kualitas dari reliability, yaitu maturity (MATU), fault tolerance (FAUL), recoverability (RECO) dan availability (AVAI). Jadi field MATU_FAUL merupakan perbandingan dari maturity dengan fault tolerance
matrik_m	tingkat preferensi dari sub faktor maintainability yaitu analysability

	(ANAL), changeability (CHAN), stability (STAB) dan testability (TEST)
matrik_fk	perbandingan tingkat preferensi dari enam faktor kualitas, yaitu functionality(F), maintainability(M), reliability(R), efficiency(E), Usability(U) dan portability(P). Jadi field FR merupakan perbandingan dari functionality dengan reliability
matrik_u	tingkat preferensi dari sub faktor kualitas usability.
bobot_akhir	bobot relatif dan absolut dari faktor dan sub faktor kualitas
kriteria	penjelasan detail dari suatu kriteria beserta perhitungan dan interpretasinya sehingga akan diperoleh suatu nilai dalam skala penilaian. Terdapat juga field rekomendasi yang menjadi acuan bagi perbaikan kualitas suatu kriteria.
penilaian_kriteria	nilai kriteria
rekomendasi	acuan untuk pemberian rekomendasi
hubungan	korelasi antara faktor kualitas, sub faktor kualitas dan kriteria. Sehingga dapat diketahui kriteria dan sub faktor dari suatu faktor kualitas.

### 3.4.2 Perancangan Alur Proses

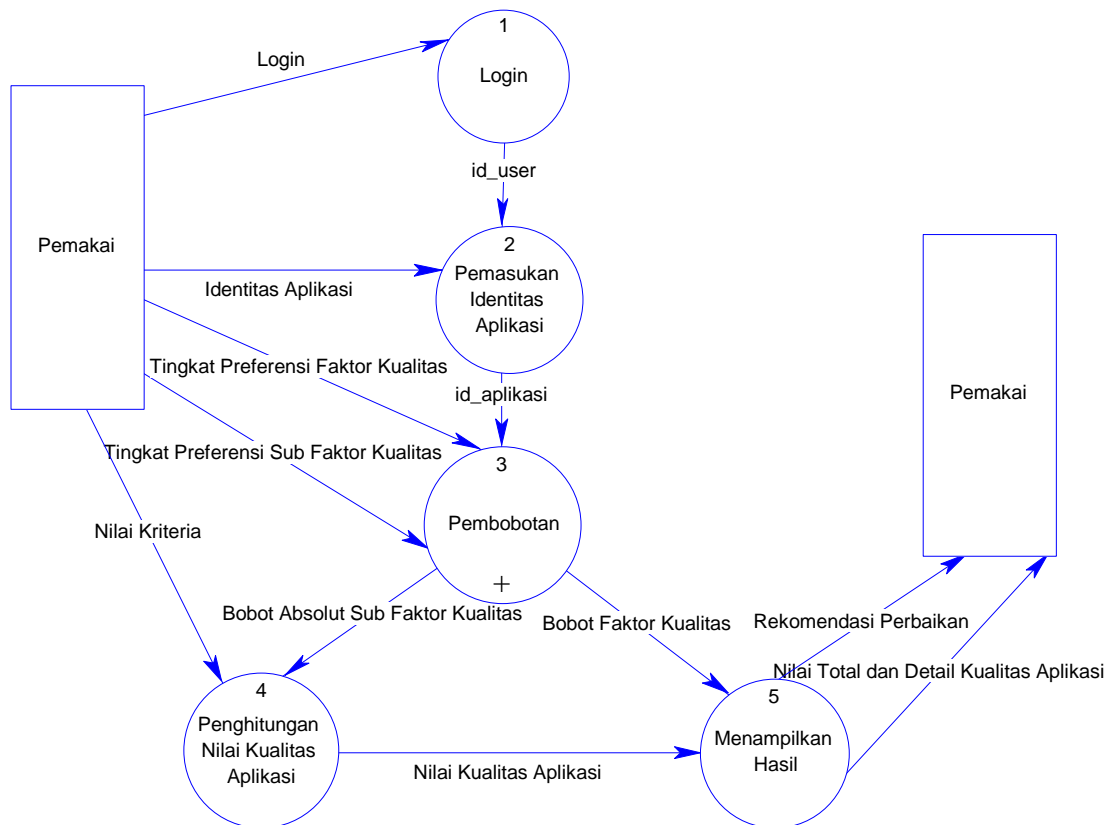
Terdapat serangkaian proses yang dijalankan pada CASE Tool ini. Untuk menjelaskannya digunakan Data Flow Diagram (DFD) yang memiliki beberapa tingkatan level tertentu sampai dihasilkan proses yang detail. Pembuatan DFD dengan menggunakan Tool Power Designer versi 6.0.

Gambar 3.4 merupakan proses global yang mencakup semua proses yang terjadi. Pemakai memasukkan semua data yang nantinya akan digunakan sebagai input dari proses yaitu: login, identitas aplikasi, tingkat preferensi faktor dan sub faktor kualitas dan nilai kriteria. Output yang dihasilkan berupa bobot relatif faktor kualitas, bobot absolut sub faktor kualitas, nilai total dan detail kualitas serta rekomendasi.



Gambar 3.4 DFD Level 0 Sistem

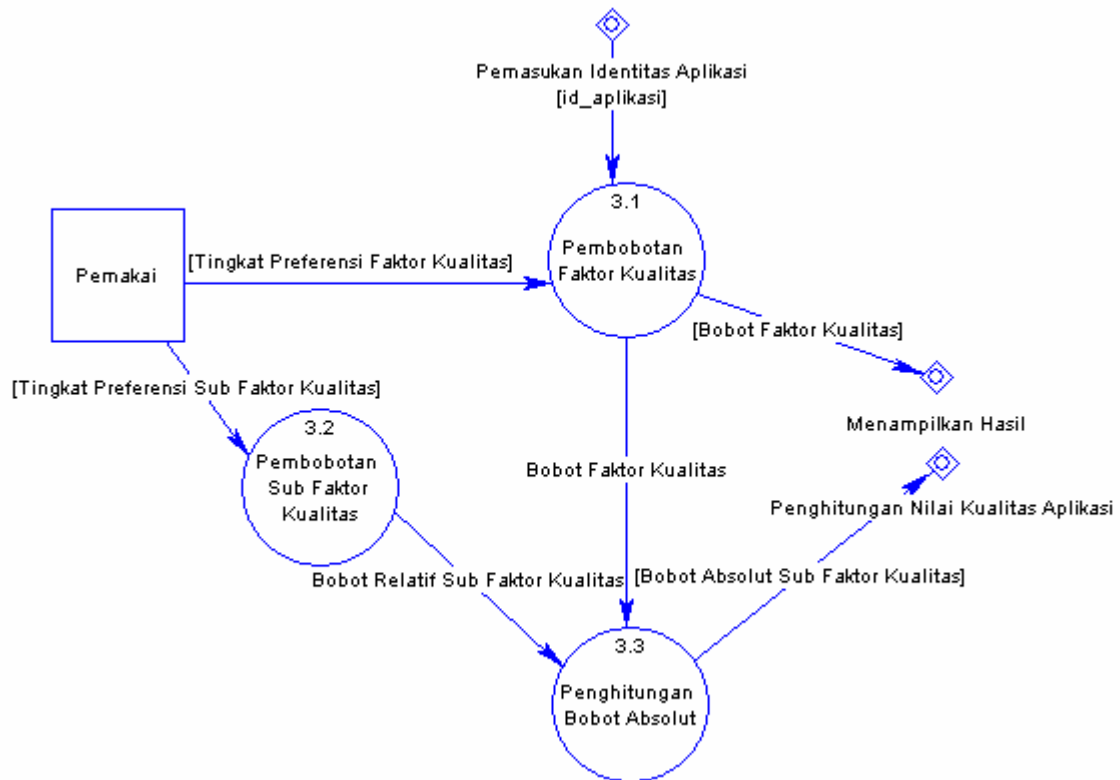
Gambar 3.5 merupakan detail dari alur data pada sistem. Sistem dibagi menjadi lima proses besar yaitu: Login, Pemasukan Identitas Aplikasi, Pembobotan, Penghitungan Nilai Kualitas dan Menampilkan Hasil.



Gambar 3.5 DFD Level 1 Detail Sistem

Proses pembobotan memiliki alur yang paling rumit, sehingga dijelaskan lebih detail pada Gambar 3.6. Proses ini terdiri dari beberapa rangkaian proses. Pertama dilakukan pembobotan faktor kualitas, kemudian pembobotan sub faktor kualitas. Hasil dari proses pembobotan faktor kualitas berupa bobot faktor kualitas dikalikan dengan bobot relatif sub faktor kualitas dari hasil proses pembobotan sub faktor kualitas menghasilkan bobot absolut. Bobot absolut digunakan untuk proses penghitungan nilai kualitas aplikasi. Sedangkan bobot faktor kualitas juga masih diperlukan untuk proses menampilkan hasil.





Gambar 3.6 DFD Level 2 Pembobotan

### 3.4.3 Perancangan Antar Muka

Pada bagian ini akan dibuat halaman masukan dari pengguna aplikasi dengan berbasis web dengan PHP. Gambar 3.7 adalah form login bagi pengguna. Tiap pengguna akan memiliki login yang unik sebelum memasukkan identitas aplikasi pada Gambar 3.8.

**Halaman Login**

User Name

Password

Reset

Login

Gambar 3.7 Form Login

FORM IDENTITAS APLIKASI	
Nama	<input type="text"/>
Versi	<input type="text"/>
Tanggal Pembuatan	<input type="text"/>
Kegunaan	<input type="text"/>
Developer	<input type="text"/>
Keterangan	<input type="text"/>
<input type="button" value="kosongkan"/> <input type="button" value="simpan"/>	

Gambar 3.8 Form Identitas Aplikasi

Untuk pembobotan terdapat dua form yaitu pembobotan faktor kualitas pada Gambar 3.9 dan pembobotan sub faktor kualitas pada Gambar 3.10.

Faktor Kualitas ke-1	Faktor Kualitas ke-2	Tingkat Preferensi
Functionality	Reliability	<input type="text"/>
Maintainability	Portability	<input type="text"/>
Usability	Efficiency	<input type="text"/>
Efficiency	Maintainability	<input type="text"/>
Reliability	Portability	<input type="text"/>
Functionality	Usability	<input type="text"/>
Reliability	Efficiency	<input type="text"/>
Efficiency	Portability	<input type="text"/>
Maintainability	Reliability	<input type="text"/>
Functionality	Efficiency	<input type="text"/>
Usability	Maintainability	<input type="text"/>
Portability	Usability	<input type="text"/>
Maintainability	Functionality	<input type="text"/>
Reliability	Usability	<input type="text"/>
Functionality	Portability	<input type="text"/>

Gambar 3.9 Form Bobot Faktor Kualitas

Faktor Kualitas	Sub-Faktor ke-1	Sub-Faktor ke-2	Tingkat Preferensi	Faktor Kualitas	Sub-Faktor ke-1	Sub-Faktor ke-2	Tingkat Preferensi
Functionality	Interoperability	Traceability		Maintainability	Analysability	Changeability	
	Accuracy	Interoperability			Changeability	Stability	
	Security	Suitability			Stability	Testability	
	Interoperability	Security			Analyzability	Stability	
	Accuracy	Suitability			Changeability	Testability	
	Suitability	Interoperability			Testability	Analyzability	
	Security	Accuracy		Usability	Understandability	Learnability	
	Suitability	Traceability			Learnability	Operability	
	Accuracy	Traceability			Operability	Attractiveness	
Portability	Traceability	Security			Understandability	Operability	
	Adaptability	Installability			Attractiveness	Understandability	
	Installability	Co-existence			Customizability	Navigability	
	Co-existence	Replaceability			Operability	Customizability	
	Adaptability	Co-existence			Learnability	Customizability	
	Installability	Replaceability			Navigability	Learnability	
Reliability	Replaceability	Adaptability			Customizability	Understandability	
	Maturity	Fault Tolerance			Learnability	Attractiveness	
	Fault Tolerance	Recoverability			Operability	Navigability	
	Recoverability	Maturity			Navigability	Understandability	

Gambar 3.10 Form Bobot Sub-Faktor Kualitas

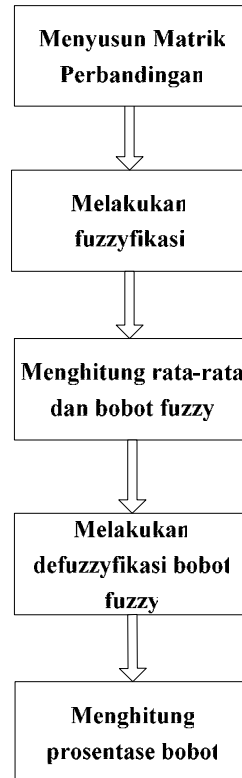
Untuk penilaian kriteria dibuat form kriteria seperti pada Gambar 3.11.

NO	METRIK	TUJUAN	KOMPUTASI	INTERPRETASI	SKALA	NILAI
1	Response time	Waktu untuk menyelesaikan sebuah tugas yang spesifik	$T=A-B$ A:Waktu untuk memperoleh hasil B:Waktu setelah perintah selesai dimasukkan	0	0: $t=0,1-0,9$ ; 1: $t=1-1,9$ ; 2: $t \geq 3,1$ ; 3: $t=2-3$	
2	Throughput	Jumlah task yang dikerjakan dengan sukses pada periode waktu yang diberikan	$X=A/B$ A:Jumlah task yang lengkap B:Periode waktu observasi	0	0: $x < 1$ ; 1: $x=1-3$ ; 2: $x=3,1-5$ ; 3: $x > 5$	
3	Turnaround time	Waktu tunggu setelah mengirimkan instruksi untuk memulai sebuah grup dari task yang berhubungan dan pelengkapanya	$X=A-B$ A:Waktu untuk memperoleh output akhir B:Waktu penyelesaian request	0	0: $t=0,1-0,9$ ; 1: $t=1-1,9$ ; 2: $t \geq 3,1$ ; 3: $t=2-3$	
4	I/O devices utilization	Waktu dari penempatan I/O devices dibagi dengan waktu khusus yang didesain untuk penempatan I/O devices	$X=A/B$ A:Waktu dari penempatan I/O devices B:waktu khusus yang didesain untuk penempatan I/O devices	$0 \leq X \leq 1$ , makin kecil dan menjauhi 1 makin baik	0: $x=0,76-1$ ; 1: $x=0,51-0,75$ ; 2: $x=0,26-0,5$ ; 3: $x=0-0,25$ ;	
5	Failure avoidance	Jumlah kesalahan pola yang dihindari pada saat desain dan coding dan dibandingkan dengan jumlah kesalahan pola yang disarankan	$X=A/B$ A:Jumlah kesalahan pola yang dapat dihindari B:Jumlah kesalahan pola yang disarankan	$0 \leq X \leq 1$ , makin mendekati 1 makin baik, user dapat lebih sering menghindari kesalahan pola	0: $x=0-0,25$ ; 1: $x=0,26-0,5$ ; 2: $x=0,51-0,75$ ; 3: $x=0,76-1$	
6	Mean recovery time	Waktu rata-rata sistem untuk menyelesaikan recovery dari recovery awal sebagian	$X=Sum(T)/B$ T:Waktu untuk recovery sistem software yang down pada tiap kesempatan B:Jumlah kasus yang diobservasi sistem software telah masuk pada recovery	$0 \leq X$ , makin kecil makin baik	0: $t > 3,1$ ; 1: $t=2-3$ ; 2: $t=1-1,9$ ; 3: $t=0-0,9$	

Gambar 3.11 Form Kriteria

#### 3.4.4 Pembobotan dengan Fuzzy AHP

Langkah-langkah pembobotan mengikuti alur pada Gambar 3.12.



Gambar 3.12 Blok Diagram Pembobotan

Dari masukan pengguna terhadap pembobotan faktor kualitas pada Gambar 3.9, dibuat matrik perbandingan dengan menggunakan persamaan (2.7). Diperoleh matrik 6x6 seperti pada Tabel 3.5 yang akan menghasilkan bobot faktor kualitas.

Tabel 3.5 Matrik Perbandingan Faktor Kualitas

	Functionality	Reliability	Maintainability	Portability	Usability	Efficiency
Functionality	1					
Reliability		1				
Maintainability			1			
Portability				1		
Usability					1	
Efficiency						1

Misal diambil contoh penghitungan bobot sub faktor kualitas dari functionality, maka didapatkan matrik perbandingan seperti pada Tabel 3.6.

Tabel 3.6 Matrik Perbandingan Sub-Faktor Kualitas

	Suitability	Accuracy	Interoperability	Security	Traceability
Suitability	1				
Accuracy		1			
Interoperability			1		
Security				1	
Traceability					1

Karena terdapat enam faktor kualitas, maka diperoleh total enam matrik perbandingan sub faktor kualitas dengan dimensinya masing-masing sesuai jumlah sub faktor kualitasnya. Matrik-matrik tersebut akan menghasilkan bobot relatif sub faktor kualitas.

Setelah didapatkan matrik perbandingan, maka dicari bilangan fuzzy dari tiap-tiap input menggunakan persamaan (2.7). Kemudian dihitung rata-rata fuzzy ( $r$ ) dengan persamaan (2.8) dan bobot fuzzy ( $w$ ) dengan persamaan (2.9). Defuzzyfikasi dilakukan terhadap bobot fuzzy ( $w$ ) dengan menggunakan persamaan (2.10) sehingga diperoleh bobot faktor kualitas dan bobot relatif sub faktor kualitas. Bobot absolut dari sub faktor kualitas diperoleh dengan mengalikan bobot faktor kualitas dengan bobot relatif sub-faktor kualitas. Hasilnya nanti akan terlihat seperti pada Tabel 3.7.

Tabel 3.7 Bobot Relatif dan Absolut

Faktor Kualitas	Bobot	Sub Faktor Kualitas	Bobot Relatif	Bobot Absolut
Functionality		Suitability		
		Accuracy		
		Interoperability		
		Security		
		Traceability		
Reliability		Fault Tolerance		
		Maturity		
		Recoverability		
		Availability		
		Understandability		
Usability		Learnability		
		Operability		
		Attractiveness		
		Customizability		
		Navigability		
Efficiency		Time Behavior		
		Resource Utilization		
Maintainability		Analysability		
		Changeability		

		Stability		
		Testability		
Portability		Adaptability		
		Installability		
		Co-existence		
		Replaceability		

Nilai kualitas total dari suatu sistem aplikasi diperoleh dari hasil perkalian bobot absolut dengan nilai kriteria masukan pengguna pada Gambar 3.11. Untuk hubungan antara sub faktor kualitas dan kriteria dapat dilihat pada Tabel 3.3. Contohnya untuk menghitung nilai kualitas dari sub faktor Adaptability, maka dilakukan perkalian antara bobot absolut Adaptability dengan nilai dari kriteria Hardware Independence, Software Independence, Hardware Environmental Adaptability dan Organizational Environmental Adaptability. Hasil dari empat nilai tersebut kemudian dijumlahkan.

Proses diatas dilakukan pada semua sub faktor kualitas yang ada. Hasil dari penjumlahan nilai kualitas dari 25 sub faktor kualitas akan berupa nilai kualitas total dari sistem.

### 3.4.5 Pembuatan Rekomendasi

Terdapat dua macam rekomendasi yang diberikan oleh CASE Tool ini.

#### 1. Rekomendasi umum terhadap keseluruhan sistem.

Dicari lima nilai kualitas terendah dari semua sub faktor kualitas setelah dilakukan pengurutan. Kemudian dicari nilai kriteria kualitas dari kelima sub faktor tersebut yang memiliki nilai  $<3$  yang berarti bukan nilai yang sempurna. Ditampilkan saran perbaikan dari tiap kriteria kualitas.

#### 2. Rekomendasi khusus untuk tiap faktor kualitas.

Dicari nilai kriteria kualitas dari tiap faktor kualitas yang memiliki nilai  $<3$ . Kemudian ditampilkan saran perbaikannya setelah diurutkan dari nilai kualitas terendah yang menunjukkan tingkat urgensi yang tinggi.

Untuk mempermudah pengguna dalam membaca nilai kualitas, diberikan grade berupa tanda bintang (\*) sebagai penanda tingkat kualitas dari faktor maupun sub faktor kualitas. Grade diperoleh dengan membandingkan nilai kualitas

berdasarkan masukan pengguna dengan nilai kualitas standar, yaitu jika semua kriteria kualitas diisi dengan nilai sempurna 3. Terdapat empat level grade, yaitu:

\*\*\* jika nilai hasil perbandingan lebih dari 66,67

\*\* jika nilai hasil perbandingan lebih dari 33,33

\* jika nilai hasil perbandingan lebih dari 0

Tanpa bintang berarti nilainya 0

### 3.5 Implementasi CASE tool

#### 3.5.1 Implementasi Data

Rancangan data yang telah dibuat sebelumnya akan diimplementasikan ke dalam lingkungan basis data MySQL. seperti ditunjukkan pada Tabel 3.8 sampai 3.11. Data isian dari tabel faktor\_kualitas dan sub\_faktor\_kualitas ada pada Tabel 3.8 dan 3.9.

Tabel 3.8 Isian Faktor Kualitas

id_fk	nama_fk	definisi_fk	ket_fk
F	Functionality	Sekumpulan atribut yang berhubungan dengan eksistensi dari sekelompok fungsi beserta propertis khususnya yang telah sesuai dengan kebutuhan	
R	Reliability	Sekumpulan atribut yang berhubungan dengan kemampuan perangkat lunak untuk merawat level performansinya pada kondisi dan periode waktu tertentu	
U	Usability	Sekumpulan atribut yang berhubungan dengan effort yang dibutuhkan untuk penggunaan dan penilaiannya secara individu pada sekelompok pengguna	
E	Efficiency	Sekumpulan atribut yang berhubungan dengan relasi antara level performansi dari perangkat lunak dengan jumlah sumber daya yang digunakan pada kondisi tertentu	
M	Maintainability	Sekumpulan atribut yang berhubungan dengan effort yang dibutuhkan untuk membuat perubahan yang khusus	
P	Portability	Sekumpulan atribut yang berhubungan dengan kemampuan perangkat lunak untuk dikirimkan dari satu lingkungan ke lingkungan lain yang berbeda	

Tabel 3.9 Isian Sub Faktor Kualitas

id_sfk	nama_sfk	definisi_sfk	ket_sfk
SUIT	Suitability	Gambaran perangkat lunak dalam menjalankan fungsi-fungsi yang ada pada perangkat lunak tersebut	
ACCU	Accuracy	Hasil yang diekspektasikan dari perangkat lunak	
FAUL	Fault Tolerance	Kemampuan perangkat lunak untuk menyesuaikan toleransi kesalahannya secara spesifik	
MATU	Maturity	Frekuensi kesalahan produk perangkat lunak yang memberikan dampak pada perangkat lunak yang digunakan sehingga kesalahan menjadi tidak nampak dan mudah dihilangkan	
INTE	Interoperability	Kemampuan perangkat lunak untuk berinteraksi dengan sistem lain	
SECU	Security	Memperlihatkan bagaimana perangkat lunak melindungi dari akses yang belum diautorisasi	
RECO	Recoverability	Kemampuan perangkat lunak untuk menetapkan kembali level performansinya dan menyelamatkan data apabila terjadi kesalahan dan kejadian tak terduga	
TRAC	Traceability	Kemampuan perangkat lunak untuk mencari kebenaran dari pemrosesan informasi pada tiap tahap proses yang berbeda	
UNDE	Understandability	Gambaran bagaimana user secara komprehensif menggunakan perangkat lunak dengan mudah	
LEAR	Learnability	Kemudahan perangkat lunak untuk dipelajari	
OPER	Operability	Kemudahan perangkat lunak untuk dioperasikan oleh pengguna	
TIME	Time Behavior	Menggambarkan seberapa cepat respon dari sistem	
RESO	Resource Utilization	Menggambarkan seberapa efisien perangkat lunak dalam menggunakan sumber daya yang ada	

Isi dari tabel kriteria ada pada Tabel 3.10.

Tabel 3.10 Isian kriteria

id_kri	metrik	tujuan	komputasi	interpretasi	skala	skala_penilaian	rekomendasi
1	Response time	Waktu untuk menyelesaikan sebuah tugas yang spesifik...	$T=A/B$ A:Waktu untuk memperoleh hasil B:Waktu se...	$0 < T$ , makin cepat makin baik	ratio	0: $t < 1$ ; 1: $1 \leq t < 2$ ; 2: $t > 3$ ; 3: $2 \leq t \leq 3$	Percepat waktu penyelesaian tugas yang spesifik
2	Throughput	Jumlah task yang dikerjakan dengan sukses pada per...	$X=A/B$ A:Jumlah task yang lengkap B:Periode waktu...	$0 < X$ , lebih besar lebih baik	ratio	0: $x < 1$ ; 1: $1 \leq x \leq 3$ ; 2: $3 < x \leq 5$ ; 3: $x > 5$	Tingkatkan jumlah task yang dapat dikerjakan denga...
3	Turnaround time	Waktu tunggu setelah mengirimkan instruksi untuk m...	$X=A/B$ A:Waktu untuk memperoleh output akhir B:Wa...	$0 < T$ , lebih pendek lebih baik	ratio	0: $0 < t < 1$ ; 1: $1 \leq t < 2$ ; 2: $t > 3$ ; 3: $2 \leq t \leq 3$	Perpendek waktu tunggu setelah pengiriman instruks...
4	I/O devices utilization	Waktu dari penempatan I/O devices dibagi dengan wa...	$X=A/B$ A:Waktu dari penempatan I/O devices B:waktu...	$0 \leq X \leq 1$ , makin kecil dan menjauhi 1 makin baik	absolut	0: $0,75 < x \leq 1$ ; 1: $0,5 < x \leq 0,75$ ; 2: $0,25 < x \leq 0,5$ ; 0,5; ...	Perpendek waktu dari penempatan I/O devices
5	Failure avoidance	Jumlah kesalahan pola yang dihindari pada saat des...	$X=A/B$ A:Jumlah kesalahan pola yang dapat dihindar...	$0 \leq X \leq 1$ , makin mendekati 1 makin baik, user da...	absolut	0: $0 \leq x \leq 0,25$ ; 1: $0,25 < x \leq 0,5$ ; 2: $0,5 < x \leq 0,75$ ; 3: ...	Tingkatkan jumlah kesalahan pola yang dapat dihind...
6	Mean recovery time	Waktu rata-rata sistem untuk menyelesaikan recover...	$X=Sum(T)/B$ T:Waktu untuk recovery sistem software...	$0 < X$ , makin kecil makin baik	ratio	0: $t > 3$ ; 1: $2 \leq t \leq 3$ ; 2: $1 \leq t < 2$ ; 3: $0 \leq t < 1$	Perpendek waktu rata-rata sistem untuk menyelesaik...
7	Restorability	Kemampuan produk untuk restoring dirinya sendiri s...	$X=A/B$ A:Jumlah kasus restore yang sukses dilakuka...	$0 \leq X \leq 1$ , makin besar dan mendekati 1 lebih baik,...	absolut	0: $0 \leq x \leq 0,25$ ; 1: $0,25 < x \leq 0,5$ ; 2: $0,5 < x \leq 0,75$ ; 3: ...	Tingkatkan kemampuan produk untuk restoring dirinya...
8	Memory utilization	Ukuran memori dari produk yang akan ditempatkan un...	Ukuran dalam byte	$0 \leq X$ makin kecil makin baik	ratio	0: $0,75 < x \leq 1$ ; 1: $0,5 < x \leq 0,75$ ; 2: $0,25 < x \leq 0,5$ ; 0,5; ...	Perkecil ukuran memori dari produk yang akan ditem...

Isi dari tabel hubungan terdapat pada Tabel 3.11.

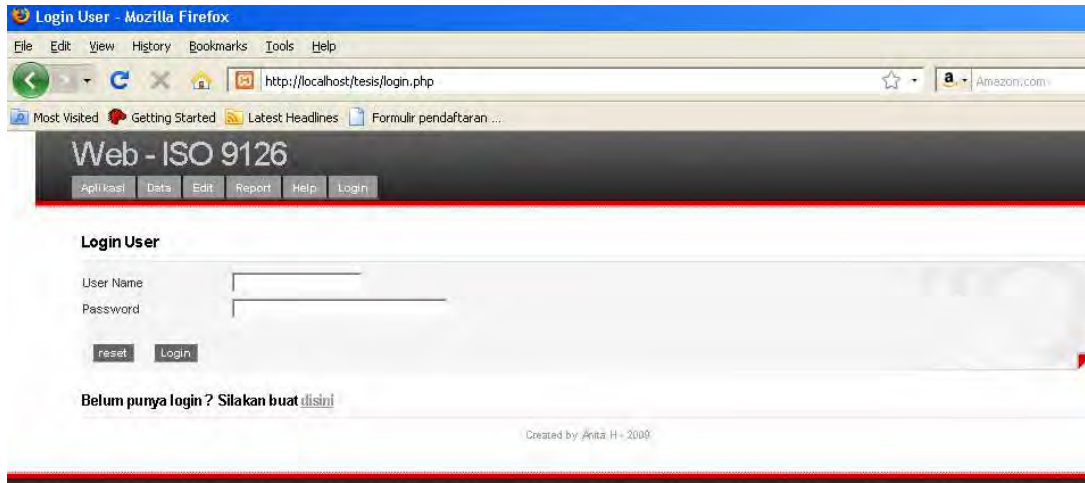
Tabel 3.11 Pemetaan Kriteria

faktor	subfaktor	kriteria
Functionality	Suitability	Completeness
Functionality	Accuracy	Computational Accuracy
Functionality	Accuracy	Correctness
Functionality	Interoperability	Communicational Commonality
Functionality	Interoperability	Uniformity
Functionality	Security	Access Audit Ability
Functionality	Security	Access Controllability
Functionality	Traceability	Audit Ability
Functionality	Traceability	Clarity
Functionality	Traceability	Data Commonality
Reliability	Fault Tolerance	Failure Avoidance
Reliability	Fault Tolerance	Incorrect Operation Avoidance
Reliability	Maturity	Robustness
Reliability	Maturity	Trust Ability
Reliability	Maturity	Degradability
Reliability	Recoverability	Mean Recovery Time
Reliability	Recoverability	Restartability
Reliability	Recoverability	Restorability
Reliability	Availability	Instrumentation
Usability	Navigability	Clarity
Usability	Understandability	Demonstration Accessibility
Usability	Learnability	Effectiveness of help system in use
Usability	Understandability	Access Audit Ability
Usability	Learnability	Effectiveness of the user documentation
Usability	Operability	Audit Ability



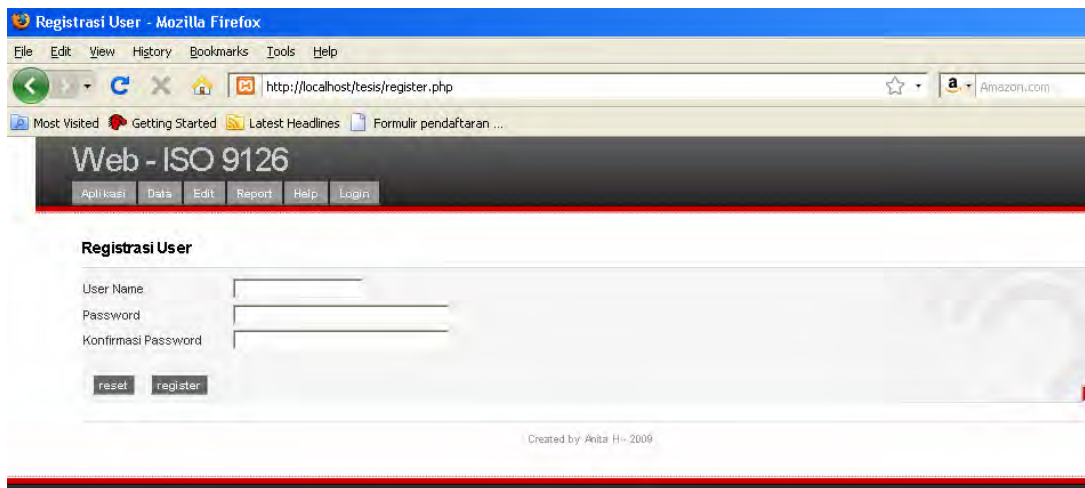
### 3.5.2 Implementasi Antar Muka

Hasil implementasi antar muka dapat dilihat pada Gambar 3.13 sampai 3.24. Halaman pertama adalah halaman login seperti pada Gambar 3.13.




Gambar 3.13 Form Jadi Login

Jika pengguna belum memiliki login, klik [disini](#) pada Gambar 3.13 sehingga muncul halaman registrasi seperti pada Gambar 3.14.

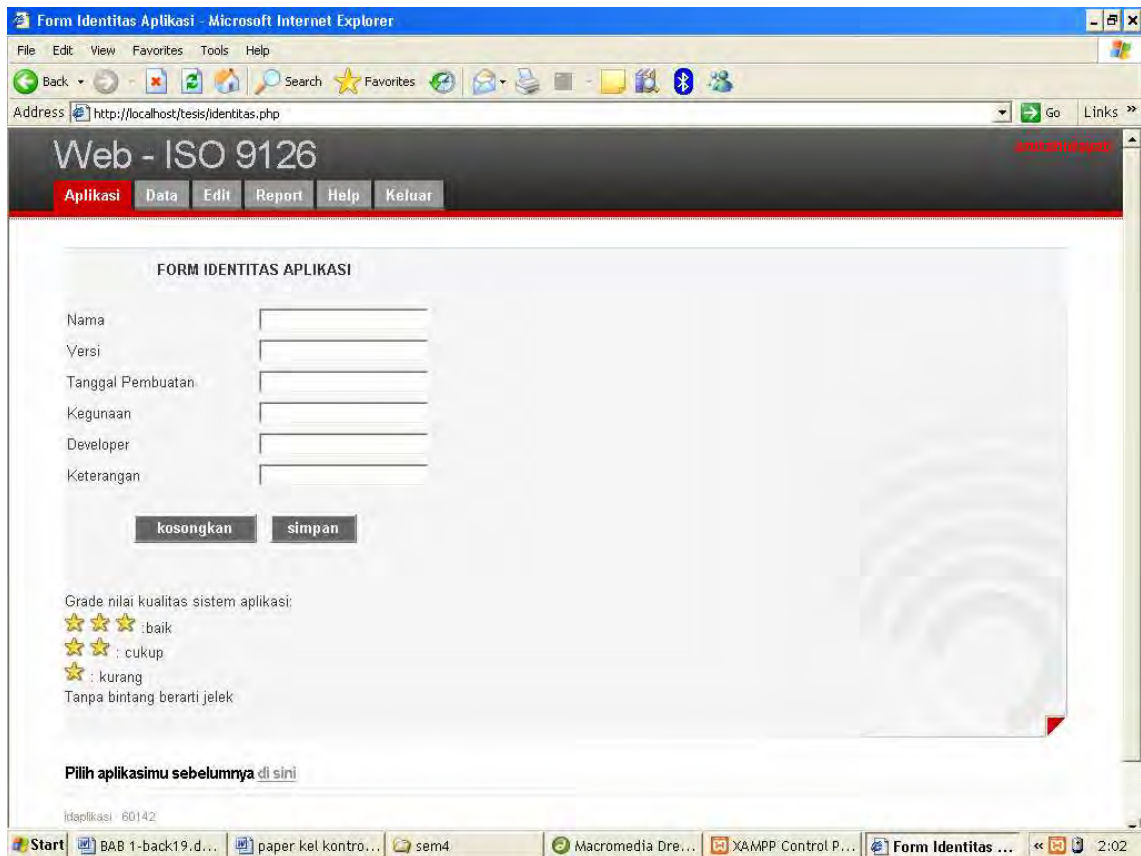


Gambar 3.14 Form Registrasi Login

Terdapat enam menu dari sistem seperti pada Gambar 3.15.

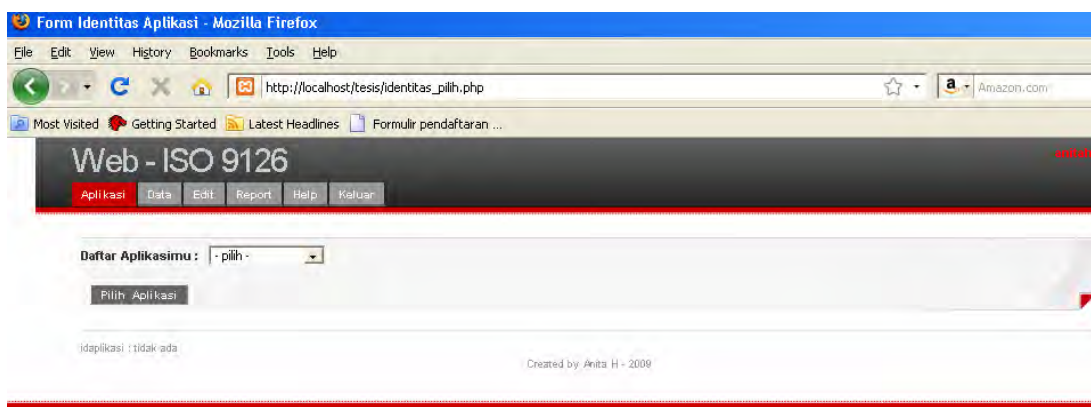


Gambar 3.15 Menu dari Aplikasi



Gambar 3.16 Form Jadi Identitas Aplikasi

Jika pengguna telah melakukan pengisian data, maka otomatis tersimpan pada user name. Data-data yang telah dimasukkan dapat dipanggil dengan klik [disini](#) pada Gambar 3.16 sehingga tampil Gambar 3.17.



Gambar 3.17 Form Pilih Aplikasi

Setelah memilih suatu aplikasi, akan keluar data aplikasi yang telah tersimpan di database. Pengguna akan diarahkan ke form Rekomendasi untuk selanjutnya dapat melakukan perubahan maupun melihat datanya.

Faktor Kualitas ke-1	Faktor Kualitas ke-2	Tingkat Preferensi
Functionality	Reliability	1
Maintainability	Portability	1
Usability	Efficiency	1
Efficiency	Maintainability	1
Reliability	Portability	1
Functionality	Usability	1
Reliability	Efficiency	1
Efficiency	Portability	1
Maintainability	Reliability	1
Functionality	Efficiency	1
Usability	Maintainability	1
Portability	Usability	1
Maintainability	Functionality	1
Reliability	Usability	1
Functionality	Portability	1

Gambar 3.18 Form Jadi Pembobotan Faktor Kualitas

Faktor Kualitas	Sub-Faktor ke-1	Sub-Faktor ke-2	Tingkat Preferensi	
Functionality	Interoperability	Traceability	1	
	Accuracy	Interoperability	1	
	Security	Suitability	1	
	Interoperability	Security	1	
	Accuracy	Suitability	1	
	Suitability	Interoperability	1	
	Security	Accuracy	1	
	Suitability	Traceability	1	
Portability	Accuracy	Traceability	1	
	Traceability	Security	1	
	Adaptability	Installability	1	
	Maintainability	Analysability	Changeability	1
		Changeability	Stability	1
		Stability	Testability	1
		Analysability	Stability	1
Changeability		Testability	1	
Testability		Analysability	1	
Usability		Understandability	Learnability	1
	Learnability	Operability	1	
	Operability	Attractiveness	1	
	Understandability	Operability	1	
	Attractiveness	Understandability	1	

Gambar 3.19 Form Jadi Pembobotan Sub Faktor Kualitas



No	Metrik	Tujuan	Komputasi	Interpretasi	Skala	Nilai
1	Response time	Waktu untuk menyelesaikan sebuah tugas yang spesifik	$T = A - B$ A: Waktu untuk memperoleh hasil B: Waktu setelah perintah selesai dimasukkan	$0 < T$ , makin cepat makin baik	$0: t \leq 1; 1: 1 < t \leq 2; 2: t > 3; 3: 2 < t \leq 3$	1
2	Throughput	Jumlah task yang dikerjakan dengan sukses pada periode waktu yang diberikan	$X = A/B$ A: Jumlah task yang lengkap B: Periode waktu observasi	$0 < X$ , lebih besar lebih baik	$0: x \leq 1; 1: 1 < x \leq 3; 2: 3 < x \leq 5; 3: x > 5$	1
3	Turnaround Time	Waktu tunggu setelah mengirimkan instruksi untuk memulai sebuah grup dari task yang berhubungan dan pelengkapannya	$X = A - B$ A: Waktu untuk memperoleh output akhir B: Waktu penyelesaian request	$0 < T$ , lebih pendek lebih baik	$0: 0 < t \leq 1; 1: 1 < t \leq 2; 2: t > 3; 3: 2 < t \leq 3$	1
4	I/O Devices Utilization	Waktu dari penempatan I/O devices dibagi dengan waktu khusus yang didesain untuk penempatan I/O devices	$X = A/B$ A: Waktu dari penempatan I/O devices B: Waktu khusus yang didesain untuk penempatan I/O devices	$0 \leq X \leq 1$ , makin kecil dan menjauhi 1 makin baik	$0: 0,75 < x \leq 1; 1: 0,5 < x \leq 0,75; 2: 0,25 < x \leq 0,5; 3: 0 < x \leq 0,25$	1
5	Failure Avoidance	Jumlah kesalahan pola yang dihindari pada saat desain dan coding dan dibandingkan dengan jumlah kesalahan pola yang disarankan	$X = A/B$ A: Jumlah kesalahan pola yang dapat dihindari B: Jumlah kesalahan pola yang disarankan	$0 \leq X \leq 1$ , makin mendekati 1 makin baik; user dapat lebih sering menghindari kesalahan pola	$0: 0 < x \leq 0,25; 1: 0,25 < x \leq 0,5; 2: 0,5 < x \leq 0,75; 3: 0,75 < x \leq 1$	1
6	Mean Recovery Time	Waktu rata-rata sistem untuk menyelesaikan recovery dari recovery awal sebagian	$X = \text{Sum}(T)/B$ T: Waktu untuk recovery sistem software yang down pada tiap kesempatan B: Jumlah kasus yang diobservasi sistem software telah masuk pada recovery	$0 \leq X$ , makin kecil makin baik	$0: t > 3; 1: 2 < t \leq 3; 2: 1 < t < 2; 3: 0 < t < 1$	1

Gambar 3.20 Form Jadi Penilaian Kriteria

Untuk definisi dari faktor, sub faktor, kriteria dan korelasi diantara ketiganya terdapat pada menu Help.

No	Faktor Kualitas	Definisi
1	Functionality	Sekumpulan atribut yang berhubungan dengan eksistensi dari sekelompok fungsi beserta propertis khususnya yang telah sesuai dengan kebutuhan
2	Reliability	Sekumpulan atribut yang berhubungan dengan kemampuan perangkat lunak untuk merawat level performansinya pada kondisi dan periode waktu tertentu
3	Usability	Sekumpulan atribut yang berhubungan dengan effort yang dibutuhkan untuk penggunaan dan penilaiannya secara individu pada sekelompok pengguna
4	Efficiency	Sekumpulan atribut yang berhubungan dengan relasi antara level performansi dari perangkat lunak dengan jumlah sumber daya yang digunakan pada kondisi tertentu
5	Maintainability	Sekumpulan atribut yang berhubungan dengan effort yang dibutuhkan untuk membuat perubahan yang khusus
6	Portability	Sekumpulan atribut yang berhubungan dengan kemampuan perangkat lunak untuk dikirinkan dari satu lingkungan ke lingkungan lain yang berbeda

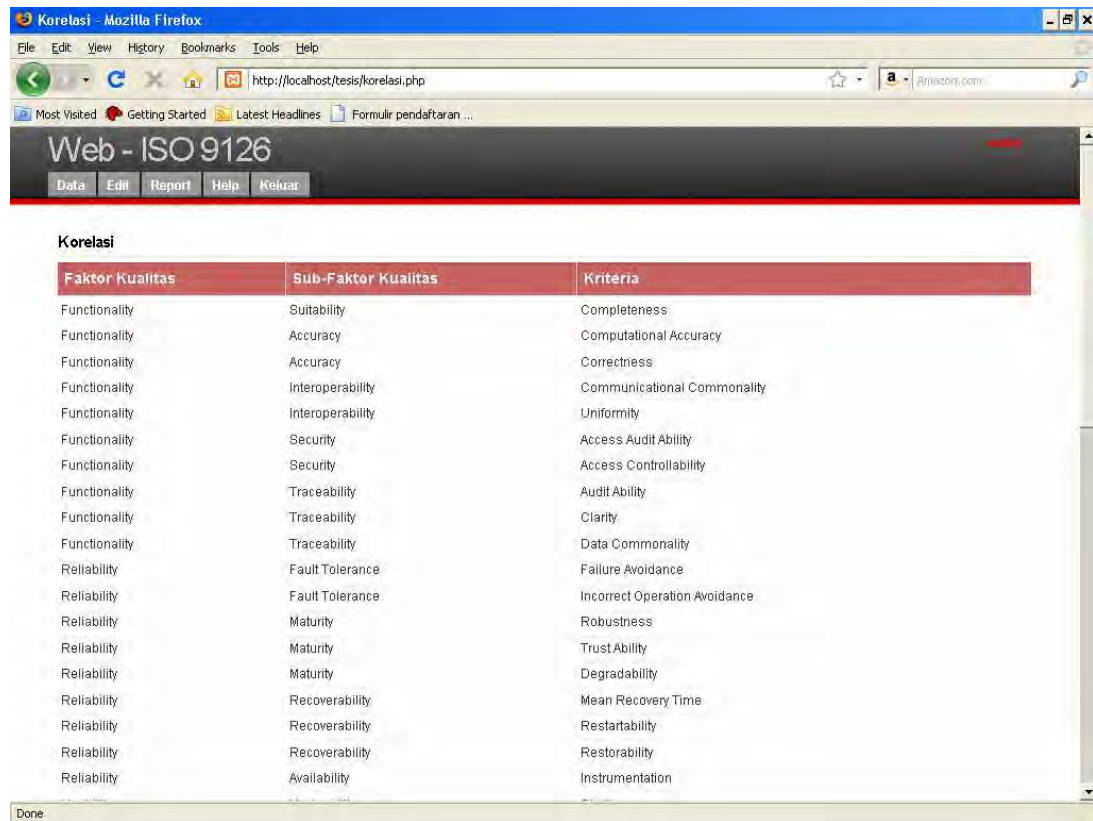
Gambar 3.21 Sub Menu Help Faktor Kualitas

No.	Sub Faktor Kualitas	Definisi
1	Suitability	Gambaran perangkat lunak dalam menjalankan fungsi-fungsi yang ada pada perangkat lunak tersebut
2	Accuracy	Hasil yang diekspektasikan dari perangkat lunak
3	Fault Tolerance	Kemampuan perangkat lunak untuk menyesuaikan toleransi kesalahannya secara spesifik
4	Maturity	Frekuensi kesalahan produk perangkat lunak yang memberikan dampak pada perangkat lunak yang digunakan sehingga kesalahan menjadi tidak nampak dan mudah dihilangkan
5	Interoperability	Kemampuan perangkat lunak untuk berinteraksi dengan sistem lain
6	Security	Memperlihatkan bagaimana perangkat lunak melindungi dari akses yang belum diautorisasi
7	Recoverability	Kemampuan perangkat lunak untuk menetapkan kembali level performansinya dan menyelamatkan data apabila terjadi kesalahan dan kejadian tak terduga
8	Traceability	Kemampuan perangkat lunak untuk mencari kebenaran dari pemrosesan informasi pada tiap tahap proses yang berbeda
9	Understandability	Gambaran bagaimana user secara komprehensif menggunakan perangkat lunak dengan mudah
10	Learnability	Kemudahan perangkat lunak untuk dipelajari
11	Operability	Kemudahan perangkat lunak untuk dioperasikan oleh pengguna
12	Time Behavior	Menggambarkan seberapa cepat respon dari sistem
13	Resource Utilization	Menggambarkan seberapa efisien perangkat lunak dalam menggunakan sumber daya yang ada
14	Analysability	Kemudahan untuk menentukan penyebab kesalahan
15	Changeability	Kemampuan untuk memodifikasi dan menghapus kesalahan dalam perubahan lingkungan
16	Stability	Merupakan toleransi dari aplikasi jika terdapat efek yang tidak diharapkan dari modifikasi
17	Testability	Kemudahan perangkat lunak dalam mengadakan uji coba terhadapnya

Gambar 3.22 Sub Menu Help Sub Faktor Kualitas

No.	Kriteria	Tujuan	Komputasi	Interpretasi	Skala	Skala Penilaian
1	Response time	Waktu untuk menyelesaikan sebuah tugas yang spesifik	$T = A - B$ A: Waktu untuk memperoleh hasil B: Waktu setelah perintah selesai dimasukkan	$0 \leq T$ , makin cepat makin baik	ratio	$0: 1 < 1; 1: 1 \leq 1 < 2; 2: 2 \leq 1 < 3$
2	Throughput	Jumlah task yang dikerjakan dengan sukses pada periode waktu yang diberikan	$X = A/B$ A: Jumlah task yang lengkap B: Periode waktu observasi	$0 \leq X$ , lebih besar lebih baik	ratio	$0: X < 1; 1: 1 \leq X < 3; 2: 3 \leq X < 5; 3: X \geq 5$
3	Turnaround time	Waktu tunggu setelah mengirimkan instruksi untuk memulai sebuah grup dari task yang berhubungan dan pelengkapannya	$X = A - B$ A: Waktu untuk memperoleh output akhir B: Waktu penyelesaian request	$0 \leq T$ , lebih pendek lebih baik	ratio	$0: 0 < 1; 1: 1 \leq 1 < 2; 2: 2 \leq 1 < 3$
4	I/O devices utilization	Waktu dari penempatan I/O devices dibagi dengan waktu khusus yang didesain untuk penempatan I/O devices	$X = A/B$ A: Waktu dari penempatan I/O devices B: waktu khusus yang didesain untuk penempatan I/O devices	$0 \leq X \leq 1$ , makin kecil dan menjauhi 1 makin baik	absolut	$0: 0,75 \leq X \leq 1; 1: 0,5 \leq X \leq 0,75; 2: 0,25 \leq X \leq 0,5; 3: 0 \leq X \leq 0,25$
5	Failure avoidance	Jumlah kesalahan pola yang dihindari pada saat desain dan coding dan dibandingkan dengan jumlah kesalahan pola yang disarankan	$X = A/B$ A: Jumlah kesalahan pola yang dapat dihindari B: Jumlah kesalahan pola yang disarankan	$0 \leq X \leq 1$ , makin mendekati 1 makin baik, user dapat lebih sering menghindari kesalahan pola	absolut	$0: 0 \leq X \leq 0,25; 1: 0,25 \leq X \leq 0,5; 2: 0,5 \leq X \leq 0,75; 3: 0,75 \leq X \leq 1$
6	Mean recovery time	Waktu rata-rata sistem untuk menyelesaikan recovery dari recovery awal sebagian	$X = \text{Sum}(T)/B$ T: Waktu untuk recovery sistem software yang down pada tiap kesempatan B: Jumlah kasus yang diobservasi	$0 \leq X$ , makin kecil makin baik	ratio	$0: 0 < 1; 1: 2 \leq 1 < 3; 2: 1 \leq 1 < 2; 3: 0 \leq 1$

Gambar 3.23 Sub Menu Help Kriteria Kualitas



Gambar 3.24 Sub Menu Help Korelasi



## BAB 4

### HASIL PENELITIAN DAN PEMBAHASAN

Pada bab ini akan dilakukan uji coba untuk mengukur kinerja dari CASE Tool yang telah dibuat. Dari hasil uji coba akan dilakukan analisa sehingga dapat diketahui tingkat keberhasilan yang telah dicapai.

#### 4.1 Uji Coba

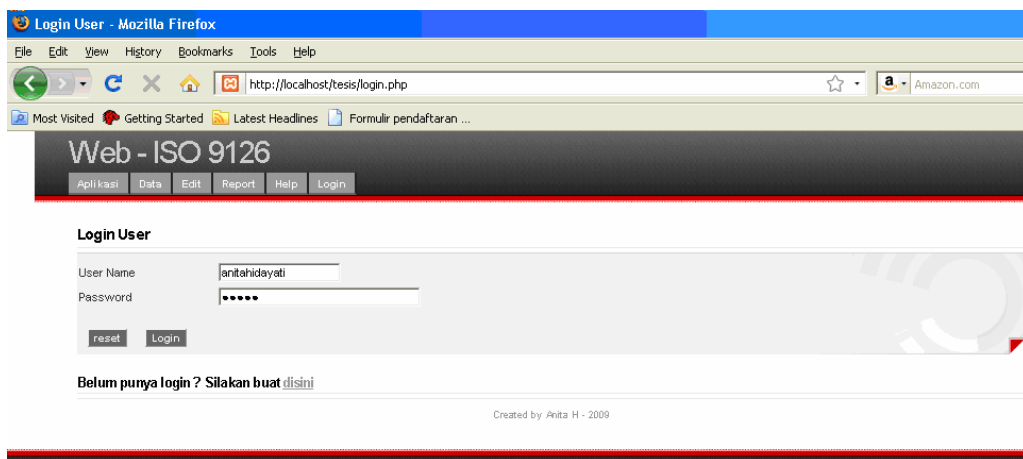
Pada uji coba yang akan dilakukan ini akan diberikan beberapa skenario untuk mengetahui fungsionalitas-fungsionalitas dari program. Selain itu, uji coba digunakan untuk menguji kesesuaian CASE Tool dengan fitur yang sudah dibuat.

##### 4.1.1 Skenario 1

Dilakukan pengisian dengan kondisi sebagai berikut:

- Perusahaan menitikberatkan pada functionality dan reliability
- Nilai dari sebagian kriteria belum sempurna, sehingga masih membutuhkan rekomendasi perbaikan

Langkah pertama pengguna memasukan username dan password pada form login seperti pada Gambar 4.1.



Gambar 4.1 Isian Form Login

Jika pengguna belum memiliki login, terlebih dahulu harus melakukan registrasi pada Gambar 4.2. User name dan password yang baru dimasukkan ke tabel users.

Tabel 4.1 Isian Tabel users

ID	username	salt	pwd	isadmin
3	admin	1247525165	2da624a6a6da8c51b948e9912ab18f0ea399dba1	1
5	anitahidayati	1247709251	f8ff570ec1f3272b8aeb092f0167efda3d8b768e	0



**Registrasi User**

User Name:

Password:

Konfirmasi Password:

Created by Anita H - 2009

Gambar 4.2 Isian Form Registrasi Login

Setelah login, pengguna memasukkan data identitas dari aplikasi yang akan dinilai kualitasnya seperti pada Gambar 4.3. Hasil isian akan dimasukkan ke tabel aplikasi seperti pada Tabel 4.2.

**FORM IDENTITAS APLIKASI**

Nama:

Versi:

Tanggal Pembuatan:

Kegunaan:

Developer:

Keterangan:

Grade nilai kualitas sistem aplikasi:

☆☆☆ : baik

☆☆ : cukup

☆ : kurang

Tanpa bintang berarti jelek

Pilih aplikasimu sebelumnya di [sini](#)

Idaplikasi: tidak ada

Created by Anita H - 2009

Gambar 4.3 Isian Form Identitas Aplikasi

Tabel 4.2 Isian Tabel aplikasi

id_aplikasi	nama_aplikasi	versi	kegunaan	tgl_pembuatan	developer	ket_aplikasi	id_user
60153	anandaceria.com	1.0.0	Berjualan busana muslim anak	1 Juli 2009	cv.antdnp	Masih tahap trial	5

Form pembobotan faktor kualitas pada Gambar 4.4 berisi penilaian tingkat preferensi dari faktor-faktor kualitas dengan skor penilaian pada Tabel 4.3.

Tabel 4.3 Skor Penilaian

Skor Penilaian	
1 – Sama Penting	
2 – Sedikit Lebih Penting	1/2 – Sedikit Tidak Lebih Penting
3 – Lebih Penting	1/3 – Tidak Lebih Penting
4 – Sangat Penting	1/4 – Sangat Tidak Penting
5 – Paling Penting	1/5 – Paling Tidak Penting

Faktor Kualitas ke-1	Faktor Kualitas ke-2	Tingkat Preferensi
Functionality	Reliability	2
Maintainability	Portability	1
Usability	Efficiency	3
Efficiency	Maintainability	2
Reliability	Portability	5
Functionality	Usability	4
Reliability	Efficiency	3
Efficiency	Portability	1
Maintainability	Reliability	2
Functionality	Efficiency	2
Usability	Maintainability	4
Portability	Usability	4
Maintainability	Functionality	3
Reliability	Usability	5
Functionality	Portability	3

Gambar 4.4 Isian Form Bobot Faktor Kualitas

Hasil penilaian tingkat preferensi dimasukkan ke tabel matrik\_fk seperti pada Tabel 4.4. Isi dari Tabel 4.4 akan dimasukkan ke matrik perbandingan seperti pada Tabel 4.5 dengan menggunakan persamaan (2.7). Kemudian dilakukan konversi ke bilangan fuzzy dengan menggunakan fungsi keanggotaan skala linguistik pada Tabel 2.4 dan 2.5.

Tabel 4.4 Isian Tabel matrik\_fk

id_mc_fk	FR	MP	UE	EM	RP	FU	RE	EP	MR	FE	UM	PU	MF	RU	FP
60153	2	1	3	2	5	4	3	1	2	2	4	4	3	5	3

Tabel 4.5 Matrik Perbandingan Faktor Kualitas

	Functionality	Reliability	Maintainability	Portability	Usability	Efficiency
Functionality	1	2	1/3	3	4	2
Reliability	1/2	1	1/2	5	5	3
Maintainability	3	2	1	1	1/4	1/2
Portability	1/3	1/5	1/1	1	4	1/1
Usability	1/4	1/5	4	1/4	1	3
Efficiency	1/2	1/3	2	1	1/3	1

Hasil konversi ke bilangan fuzzy terdapat pada Tabel 4.6.

Tabel 4.6 Konversi Faktor Kualitas ke Bilangan Fuzzy

	Functionality	Reliability	Maintainability	Portability	Usability	Efficiency
Functionality	1	(1,3,5)	(1/7,1/5,1/3)	(3,5,7)	(5,7,9)	(1,3,5)
Reliability	(1/5,1/3,1)	1	(1/5,1/3,1)	(7,9,9)	(7,9,9)	(3,5,7)
Maintainability	(3,5,7)	(1,3,5)	1	(1,1,3)	(1/9,1/7,1/5)	(1/5,1/3,1)
Portability	(1/7,1/5,1/3)	(1/9,1/9,1/7)	(1/3,1,1)	1	(5,7,9)	(1/3,1,1)
Usability	(1/9,1/7,1/5)	(1/9,1/9,1/7)	(5,7,9)	(1/9,1/7,1/5)	1	(3,5,7)
Efficiency	(1/5,1/3,1)	(1/7,1/5,1/3)	(1,3,5)	(1,1,3)	(1/7,1/5,1/3)	1

Langkah-langkah pengerjaan dari matrik perbandingan sampai dengan diperoleh bobot akhir terdapat pada Gambar 4.5.

Microsoft Excel - tesis.xls

File Edit View Insert Format Tools Data Window Help

Type a question for help

Times New Roman 10 B I U

L15

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Dicari rata-rata dengan menggunakan persamaan (2.8) sehingga menghasilkan nilai rata-rata:													
2		Lower	Middle	Upper										
3	Functionality	2,142857143	63	525										
4	Reliability	5,88	45	567										
5	Maintainability	0,066666667	0,7142857	21										
6	Portability	0,008818342	0,1555556	0,428571429										
7	Usability	0,020576132	0,0793651	0,36										
8	Efficiency	0,004081633	0,04	1,666666667										
9														
10	Dicari bobot dengan menggunakan persamaan (2.9) sehingga menghasilkan nilai:													
11		Lower	Middle	Upper										
12	Functionality	0,001921061	0,5780389	64,63129453										
13	Reliability	0,00527139	0,4128849	69,80179809										
14	Maintainability	5,97663E-05	0,0065337	2,585251781										
15	Portability	7,9056E-06	0,0014273	0,05276024										
16	Usability	1,84464E-05	0,0007282	0,044318602										
17	Efficiency	3,63916E-06	0,000367	0,205178713										
18														
19	Dilakukan defuzzifikasi terhadap bobot fuzzy diatas dengan menggunakan persamaan (2.10):													
20	Dihasilkan nilai bobot faktor kualitas berikut:													
21	Functionality	:	21,737085											
22	Reliability	:	23,406651											
23	Maintainability	:	0,8639551											
24	Portability	:	0,0180651											
25	Usability	:	0,0150217											
26	Efficiency	:	0,0685165											
27														
28	Dari bobot diatas dicari prosentase tiap-tiap faktor kualitas sehingga menghasilkan bobot akhir:													
29	Functionality	:	47,142523											
30	Reliability	:	50,763412											
31	Maintainability	:	1,8737114											
32	Portability	:	0,0391789											
33	Usability	:	0,0325786											
34	Efficiency	:	0,1483958											
35														

Sheet1 / Sheet2 / Sheet3 /

Draw AutoShapes

Ready

Gambar 4.5 Langkah Pembobotan di Ms.Excel

Untuk mendapatkan bobot dari sub faktor kualitas, pengguna harus mengisi form pada Gambar 4.6 yang berupa isian tingkat preferensi dari sub faktor kualitas dari keenam faktor kualitas. Skor penilaiannya sama dengan skor sewaktu penilaian faktor kualitas.

**Form Pembobotan Sub-Faktor Kualitas**

Penilaian dilakukan dengan membandingkan tingkat kepentingan Sub-Faktor Kualitas ke-1 terhadap Sub-Faktor Kualitas ke-2.

Isi kotak dengan skor penilaian di tabel bawah.

Faktor Kualitas	Sub-Faktor ke-1	Sub-Faktor ke-2	Tingkat Preferensi	Faktor Kualitas	Sub-Faktor ke-1	Sub-Faktor ke-2	Tingkat Preferensi	
Functionality	Interoperability	Traceability	3	Maintainability	Analyzability	Changeability	2	
	Accuracy	Interoperability	2		Changeability	Stability	2	
	Security	Suitability	5		Stability	Testability	4	
	Interoperability	Security	3		Analyzability	Stability	2	
	Accuracy	Suitability	2		Changeability	Testability	2	
	Suitability	Interoperability	3		Testability	Analyzability	4	
	Security	Accuracy	4		Usability	Understandability	Learnability	3
	Suitability	Traceability	2			Learnability	Operability	2
	Accuracy	Traceability	2			Operability	Attractiveness	3
	Traceability	Security	3			Understandability	Operability	4
Portability	Adaptability	Installability	3	Attractiveness		Understandability	3	
	Installability	Co-existence	2	Customizability		Navigability	2	
	Co-existence	Replaceability	3	Operability		Customizability	4	
	Adaptability	Co-existence	3	Learnability		Customizability	3	

Gambar 4.6 Isian Form Bobot Sub Faktor Kualitas

Hasil isian pengguna pada Gambar 4.6 akan dimasukkan ke enam tabel dari sub faktor kualitas yaitu Tabel 4.7 sampai 4.12.

Tabel 4.7 Isian Matrik Efficiency

id_mc_e	TIME_RESO
60153	2

Tabel 4.8 Isian Matrik Functionality

id_mc_f	INTE_TRAC	ACCU_INTE	SECU_SUIT	INTE_SECU	ACCU_SUIT	SUIT_INTE	SECU_ACCU	SUIT_TRAC	ACCU_TRAC	TRAC_SECU
60153	1	3	5	3	3	3	1	3	4	2

Tabel 4.9 Isian Matrik Maintainability

id_mc_m	ANAL_CHAN	CHAN_STAB	STAB_TEST	ANAL_STAB	CHAN_TEST	TEST_ANAL
60153	2	4	4	2	4	2

Tabel 4.10 Isian Matrik Portability

id_mc_p	ADAP_INST	INST_COEX	COEX_REPL	ADAP_COEX	INST_REPL	REPL_ADAP
60153	1	3	5	3	1	3

Tabel 4.11 Isian Matrik Reliability

id_mc_r	MATU_FAUL	FAUL_RECO	RECO_MATU	FAUL_AVAI	AVAI_MATU	RECO_AVAI
60153	5	3	1	3	5	3

Tabel 4.12 Isian Matrik Usability

id_mc_u	UNDE_LEAR	LEAR_OPER	OPER_ATTR	UNDE_OPER	ATTR_UNDE	CUST_NAVI	OPER_CUST	LEAR_CUST	NAVI_LEAR
60153	2	4	4	2	2	4	4	2	1

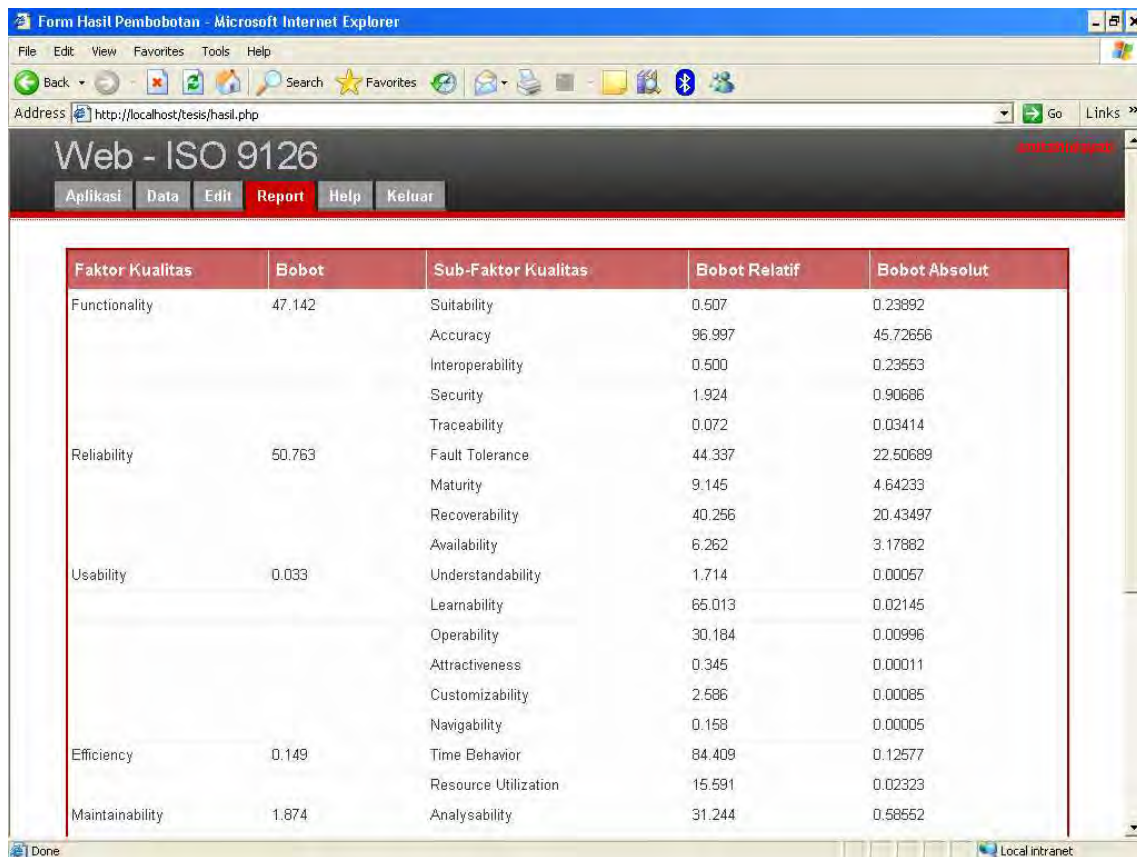
Pada tiap-tiap tabel akan dilakukan proses pembobotan yang sama seperti yang dilakukan pada faktor kualitas. Dimulai dengan konversi ke bilangan fuzzy, mencari rata-rata, bobot, defuzzyfikasi sampai prosentase bobot sub faktor kualitas. Hasil dari proses tersebut berupa bobot relatif. Bobot absolut diperoleh dengan mengalikan bobot relatif dari sub faktor kualitas dengan bobot akhir dari faktor kualitas. Bobot relatif dan absolut terdapat pada tabel bobot seperti Tabel 4.13.

Tabel 4.13 Hasil Bobot

id	faktor_subfaktor	bobot_relatif	bobot_absolut
60153	Functionality	47.142	0.00000
60153	Reliability	50.763	0.00000
60153	Usability	0.033	0.00000
60153	Efficiency	0.149	0.00000
60153	Maintainability	1.874	0.00000
60153	Portability	0.039	0.00000
60153	Suitability	0.507	0.23892
60153	Accuracy	96.997	45.72656
60153	Security	1.924	0.90686
60153	Interoperability	0.500	0.23553
60153	Traceability	0.072	0.03414
60153	Fault Tolerance	44.337	22.50689
60153	Maturity	9.145	4.64233
60153	Recoverability	40.256	20.43497
60153	Availability	6.262	3.17882
60153	Understandability	1.714	0.00057
60153	Learnability	65.013	0.02145
60153	Operability	30.184	0.00996
60153	Attractiveness	0.345	0.00011
60153	Customizability	2.586	0.00085
60153	Navigability	0.158	0.00005
60153	Time Behavior	84.409	0.12577
60153	Resource Utilization	15.591	0.02323
60153	Analysability	31.244	0.58552
60153	Changeability	65.645	1.23019
60153	Stability	2.714	0.05086
60153	Testability	0.397	0.00743
60153	Adaptability	22.851	0.00891
60153	Co-Existence	3.194	0.00125
60153	Installability	70.307	0.02742
60153	Replaceability	3.648	0.00142



Bobot dari faktor dan sub faktor kualitas ditampilkan pada form pembobotan seperti pada Gambar 4.7.



Faktor Kualitas	Bobot	Sub-Faktor Kualitas	Bobot Relatif	Bobot Absolut
Functionality	47.142	Suitability	0.507	0.23892
		Accuracy	96.997	45.72656
		Interoperability	0.500	0.23553
		Security	1.924	0.90686
		Traceability	0.072	0.03414
Reliability	50.763	Fault Tolerance	44.337	22.50689
		Maturity	9.145	4.64233
		Recoverability	40.256	20.43497
		Availability	6.262	3.17882
Usability	0.033	Understandability	1.714	0.00057
		Learnability	65.013	0.02145
		Operability	30.184	0.00996
		Attractiveness	0.345	0.00011
		Customizability	2.586	0.00085
Efficiency	0.149	Navigability	0.158	0.00005
		Time Behavior	84.409	0.12577
		Resource Utilization	15.591	0.02323
Maintainability	1.874	Analysability	31.244	0.58552

Gambar 4.7 Tampilan Bobot Faktor dan Sub Faktor Kualitas

Setelah pengisian dua form pembobotan diatas, pengguna diharuskan mengisi form penilaian kriteria pada Gambar 4.8. Tiap kriteria memiliki komputasi yang jika dimasukkan nilai variabelnya akan menghasilkan nilai yang berada dalam skala penilaian 0 sampai 3.

Form Penilaian Kriteria - Microsoft Internet Explorer

Address http://localhost/tesis/kriteria.php?idaplikasi=60153

## Web - ISO 9126

Aplikasi Data Edit Report Help Keluar

### Form Penilaian Kriteria

No	Metrik	Tujuan	Komputasi	Interpretasi	Skala	Nilai
1	Response time	Waktu untuk menyelesaikan sebuah tugas yang spesifik	$T=A-B$ A:Waktu untuk memperoleh hasil B:Waktu setelah perintah selesai dimasukkan	$D<T$ , makin cepat makin baik	0: $t<1$ ; 1: $1<t<2$ ; 2: $t>3$ ; 3: $2<t<=3$	2
2	Throughput	Jumlah task yang dikerjakan dengan sukses pada periode waktu yang diberikan	$X=A/B$ A:Jumlah task yang lengkap B:Periode waktu observasi	$D<X$ , lebih besar lebih baik	0: $x<1$ ; 1: $1<=x<=3$ ; 2: $3<x<=5$ ; 3: $x>5$	3
3	Turnaround Time	Waktu tunggu setelah mengirimkan instruksi untuk memulai sebuah grup dari task yang berhubungan dan pelengkapanya	$X=A-B$ A:Waktu untuk memperoleh output akhir B:Waktu penyelesaian request	$D<T$ , lebih pendek lebih baik	0: $0<t<1$ ; 1: $1<=t<2$ ; 2: $t>3$ ; 3: $2<=t<=3$	1
4	I/O Devices Utilization	Waktu dari penempatan I/O devices dibagi dengan waktu khusus yang didesain untuk penempatan I/O devices	$X=A/B$ A:Waktu dari penempatan I/O devices B:waktu khusus yang didesain untuk penempatan I/O devices	$D<=X<=1$ , makin kecil dan menjauhi 1 makin baik	0: $0,75<x<=1$ ; 1: $0,5<x<=0,75$ ; 2: $0,25<x<=0,5$ ; 3: $0<=x<=0,25$	3
5	Failure Avoidance	Jumlah kesalahan pola yang dihindari pada saat desain dan coding dan dibandingkan dengan jumlah kesalahan pola yang disarankan	$X=A/B$ A:Jumlah kesalahan pola yang dapat dihindari B:Jumlah kesalahan pola yang disarankan	$D<=X<=1$ , makin mendekati 1 makin baik, user dapat lebih sering menghindari kesalahan pola	0: $0<=x<=0,25$ ; 1: $0,25<x<=0,5$ ; 2: $0,5<x<=0,75$ ; 3: $0,75<x<=1$	2

$X=Sum(T)/B$  T:Waktu untuk recovery

Gambar 4.8 Isian Form Kriteria

Data isian pada Gambar 4.8 dimasukkan ke dalam tabel penilaian kriteria seperti pada Tabel 4.14.

Tabel 4.14 Isian Penilaian Kriteria

id_user	id_kriteria	nilai
60153	Response time	2
60153	Throughput	3
60153	Turnaround Time	1
60153	I/O Devices Utilization	3
60153	Failure Avoidance	2
60153	Mean Recovery Time	1
60153	Restorability	2
60153	Memory Utilization	3
60153	Incorrect Operation Avoidance	2
60153	Restartability	3
60153	Modularity	2
60153	Computational Accuracy	2
60153	Access Audit Ability	1
60153	Access Controllability	2
60153	Completeness of Description	3
60153	Modification Complexity	2
60153	Globalization	3
60153	Ease of Installation	3
60153	Demonstration Accesibility	2
60153	Effectiveness of The User Documentation	1
60153	Effectiveness of Help System in Use	2
60153	Audit Ability	1
60153	Audit Trail Capability	2
60153	Clarity	2
60153	Communicational Commonality	2
60153	Completeness	3
60153	Hardware Independence	2

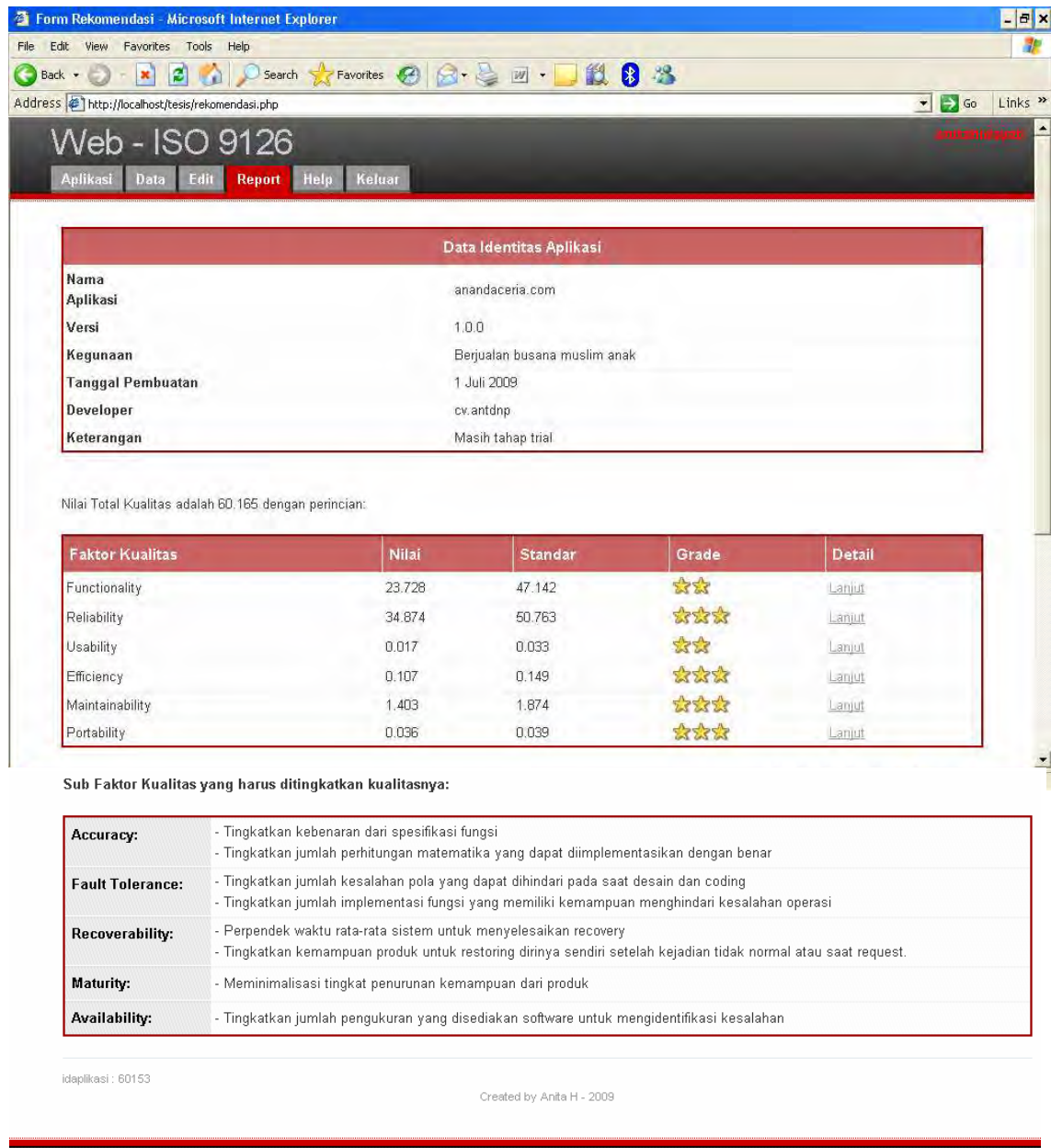


Nilai kriteria dari masukan pengguna dikalikan dengan bobot absolut pada Gambar 4.8 akan menghasilkan nilai total kualitas dari sistem. Nilai kualitas dari tiap faktor kualitas dimasukkan dalam Tabel 4.15.

Tabel 4.15 Hasil Rekomendasi

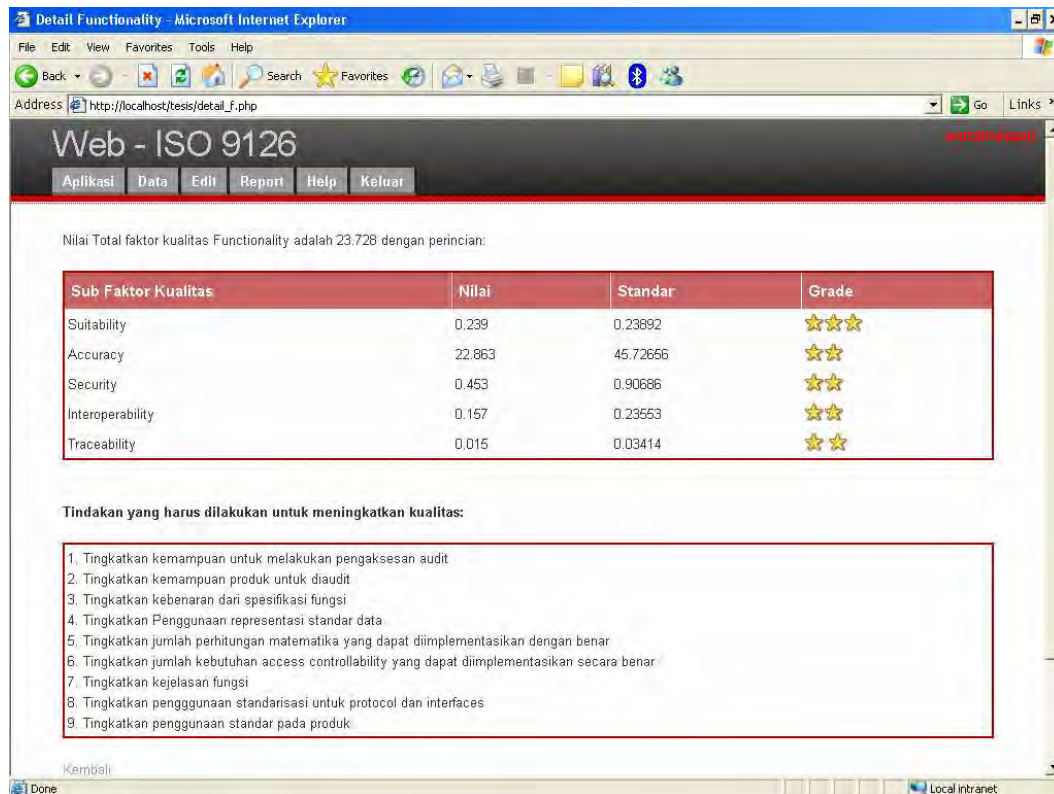
<b>id</b>	<b>faktor</b>	<b>nilai</b>
60153	Functionality	23.728
60153	Reliability	34.874
60153	Portability	0.036
60153	Usability	0.017
60153	Efficiency	0.107
60153	Maintainability	1.403
60153	Suitability	0.239
60153	Accuracy	22.863
60153	Security	0.453
60153	Interoperability	0.157
60153	Traceability	0.015
60153	Fault Tolerance	15.005
60153	Maturity	4.127
60153	Recoverability	13.623
60153	Availability	2.119
60153	Understandability	0.000
60153	Learnability	0.011
60153	Operability	0.005
60153	Customizability	0.001
60153	Navigability	0.000
60153	Attractiveness	0.000
60153	Time Behavior	0.084
60153	Resource Utilization	0.023
60153	Analysability	0.390
60153	Changeability	0.957
60153	Stability	0.051
60153	Testability	0.005
60153	Adaptability	0.007

Dari penjumlahan nilai semua faktor kualitas, diperoleh nilai kualitas dari sistem seperti pada Gambar 4.9.

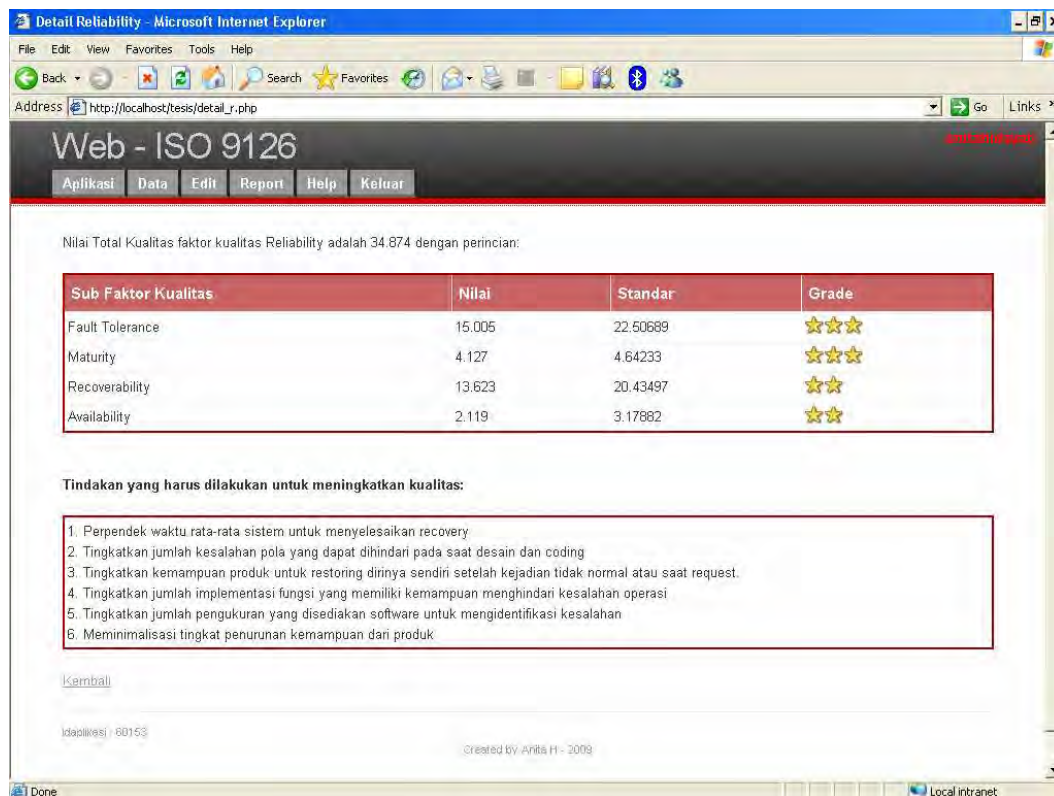


Gambar 4.9 Rekomendasi Keseluruhan

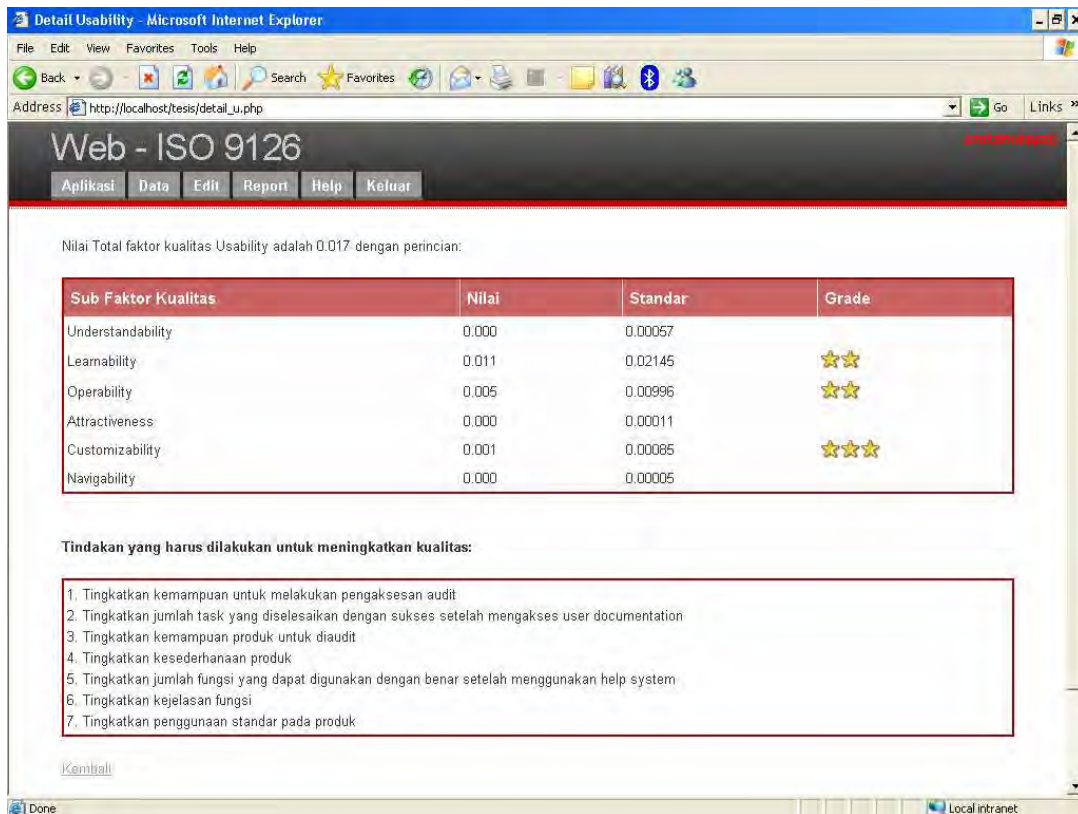
Dari rekomendasi secara global, diberikan detail untuk tiap faktor kualitas seperti pada Gambar 4.10 sampai dengan 4.15.



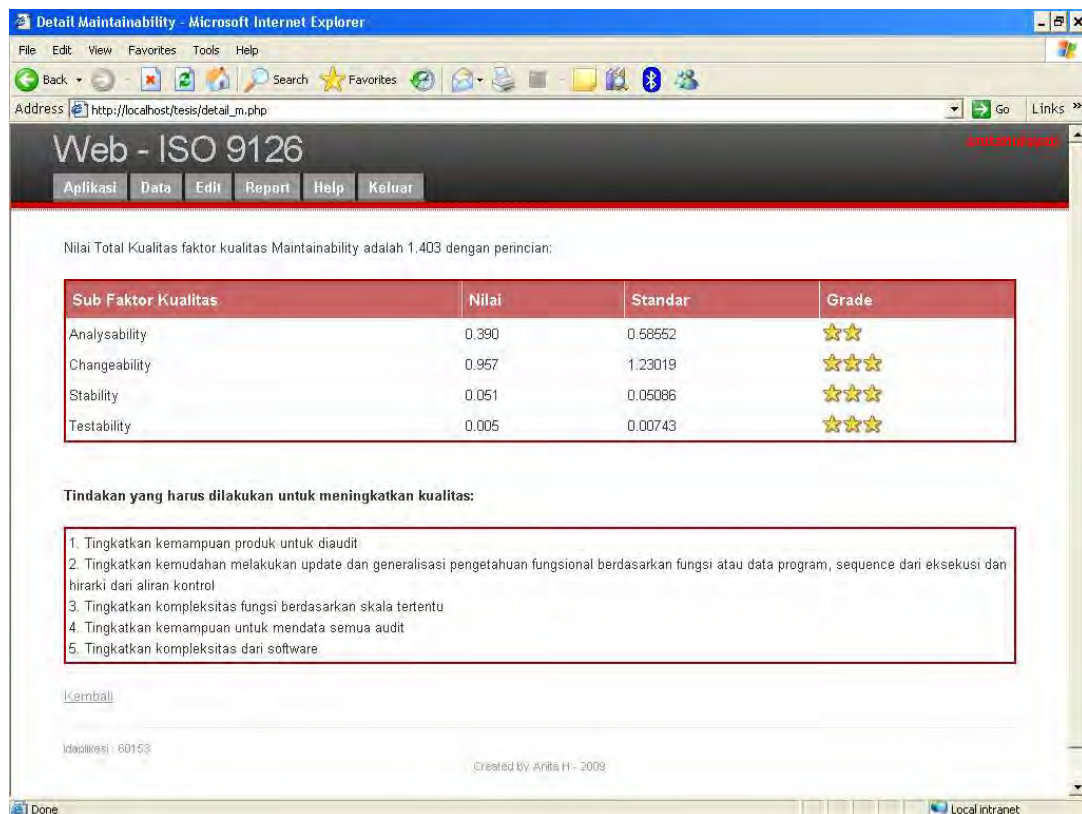
Gambar 4.10 Rekomendasi Functionality



Gambar 4.11 Rekomendasi Reliability

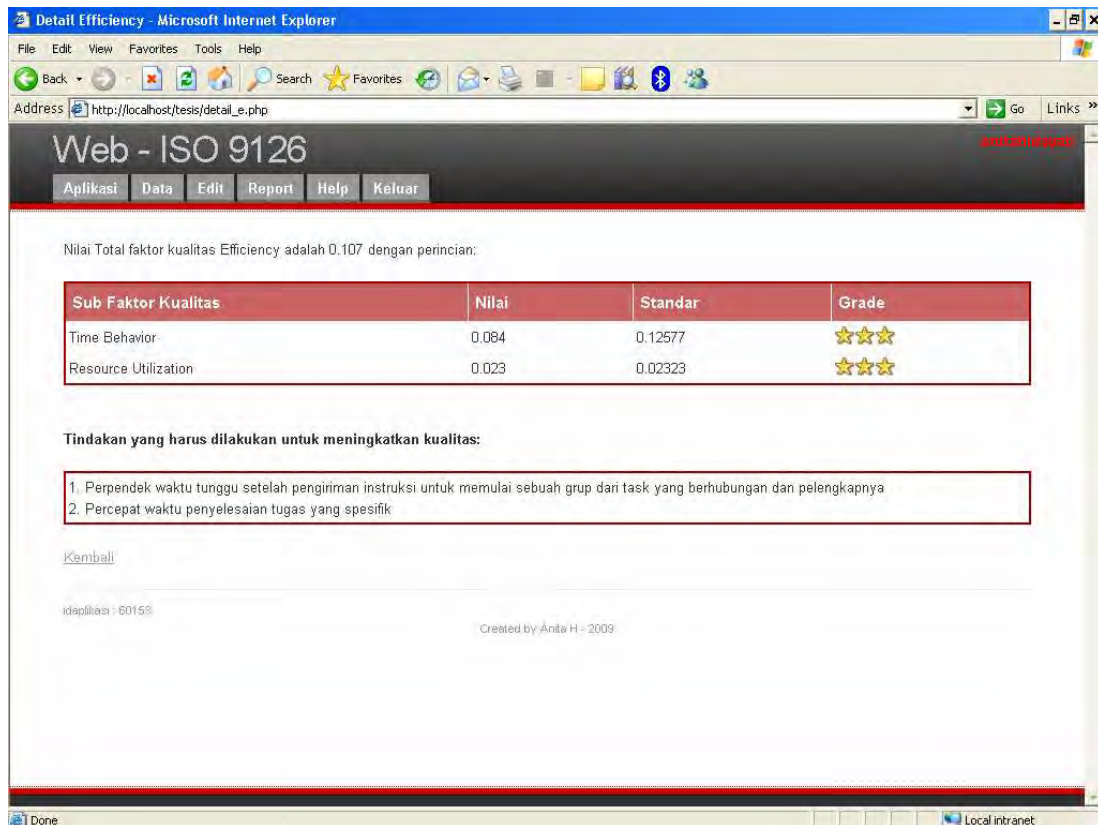


Gambar 4.12 Rekomendasi Usability

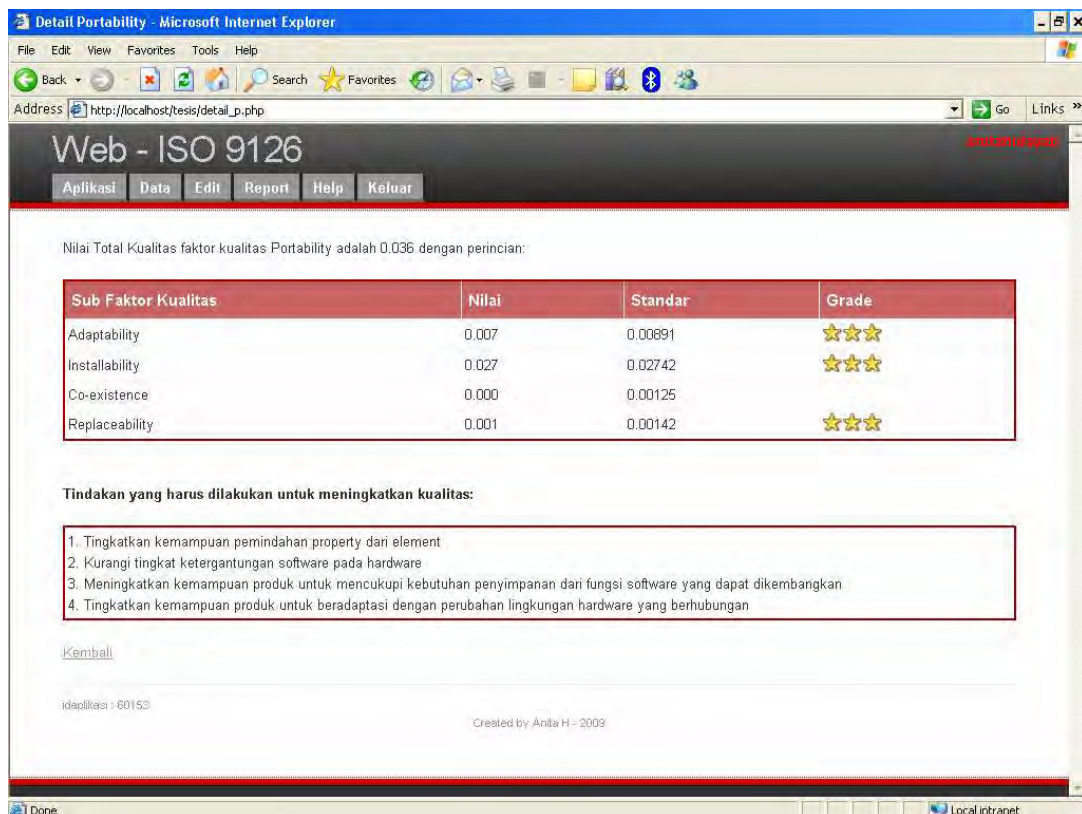


Gambar 4.13 Rekomendasi Maintainability



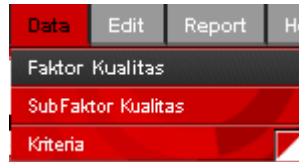


Gambar 4.14 Rekomendasi Efficiency



Gambar 4.15 Rekomendasi Portability

Untuk melihat data yang telah dimasukkan, dengan memilih menu Data. Terdapat tiga sub menu yaitu: faktor kualitas, sub faktor kualitas dan kriteria seperti pada Gambar 4.16.



Gambar 4.16 Menu dan Sub Menu Data

Tiap-tiap form dari sub menu Data terdapat pada Gambar 4.17 sampai 4.19.

Web - ISO 9126

Applikasi Data Edit Report Help Keluar

**Data Pembobotan Faktor Kualitas**

Faktor Kualitas ke-1	Faktor Kualitas ke-2	Tingkat Preferensi
Functionality	Reliability	2
Maintainability	Portability	1
Usability	Efficiency	3
Efficiency	Maintainability	2
Reliability	Portability	5
Functionality	Usability	4
Reliability	Efficiency	3
Efficiency	Portability	1
Maintainability	Reliability	2
Functionality	Efficiency	2
Usability	Maintainability	4
Portability	Usability	4
Maintainability	Functionality	3
Reliability	Usability	5
Functionality	Portability	5

Idaplikasi : 60142

Created by Anita H - 2009

Done

Gambar 4.17 Sub Menu Data Faktor Kualitas

Data Input Sub Faktor Kualitas - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost/tesis/subfaktorkualitas\_view.php

Most Visited Getting Started Latest Headlines Formulir pendaftaran ...

# Web - ISO 9126

Applikasi Data Edit Report Help Keluar

## Data Pembobotan Sub Faktor Kualitas

Faktor Kualitas	Sub-Faktor ke-1	Sub-Faktor ke-2	Tingkat Preferensi	Faktor Kualitas	Sub-Faktor ke-1	Sub-Faktor ke-2	Tingkat Preferensi
Functionality	Interoperability	Traceability	3	Maintainability	Analysability	Changeability	2
	Accuracy	Interoperability	2		Changeability	Stability	2
	Security	Suitability	5		Stability	Testability	4
	Interoperability	Security	3		Analysability	Stability	2
	Accuracy	Suitability	2	Changeability	Testability	2	
	Suitability	Interoperability	3	Testability	Analysability	4	
	Security	Accuracy	4	Usability	Understandability	Learnability	3
	Suitability	Traceability	2		Learnability	Operability	2
Accuracy	Traceability	2	Operability		Attractiveness	3	
Traceability	Security	3	Understandability		Operability	4	
Portability	Adaptability	Installability	3	Attractiveness	Understandability	3	
	Installability	Co-existence	2	Customizability	Navigability	2	
	Co-existence	Replaceability	3	Operability	Customizability	4	
	Adaptability	Co-existence	3	Learnability	Customizability	3	
	Installability	Replaceability	2	Navigability	Learnability	3	
	Replaceability	Adaptability	2	Customizability	Understandability	3	
Reliability	Maturity	Fault Tolerance	2	Learnability	Attractiveness	4	

Done

Gambar 4.18 Sub Menu Data Sub Faktor Kualitas

**Data Input Kriteria - Mozilla Firefox**

File Edit View History Bookmarks Tools Help

http://localhost/tesis/kriteria\_view.php

Most Visited Getting Started Latest Headlines Formulir pendaftaran ...

## Web - ISO 9126

Aplicasi Data Edit Report Help Keluar

### Data Penilaian Kriteria Kualitas

No.	Metrik	Tujuan	Komputasi	Interpretasi	Skala	Nilai
1	Response time	Waktu untuk menyelesaikan sebuah tugas yang spesifik	$T = A - B$ A: Waktu untuk memperoleh hasil B: Waktu setelah perintah selesai dimasukkan	$0 < T$ , makin cepat makin baik	0: $1 < 1$ ; 1: $1 < t < 2$ ; 2: $t > 3$ ; 3: $2 < t < 3$	1
2	Throughput	Jumlah task yang dikerjakan dengan sukses pada periode waktu yang diberikan	$X = A/B$ A: Jumlah task yang lengkap B: Periode waktu observasi	$0 < X$ , lebih besar lebih baik	0: $x < 1$ ; 1: $1 < x < 3$ ; 2: $3 < x < 5$ ; 3: $x > 5$	2
3	Turnaround Time	Waktu tunggu setelah mengirimkan instruksi untuk memul sebuah grup dari task yang berhubungan dan pelengkapannya	$X = A - B$ A: Waktu untuk memperoleh output akhir B: Waktu penyelesaian request	$0 < T$ , lebih pendek lebih baik	0: $0 < t < 1$ ; 1: $1 < t < 2$ ; 2: $t > 3$ ; 3: $2 < t < 3$	3
4	I/O Devices Utilization	Waktu dari penempatan I/O devices dibagi dengan waktu khusus yang didesain untuk penempatan I/O devices	$X = A/B$ A: Waktu dari penempatan I/O devices B: waktu khusus yang didesain untuk penempatan I/O devices	$0 < x < 1$ , makin kecil dan menjauhi 1 makin baik	0: $0,75 < x < 1$ ; 1: $0,5 < x < 0,75$ ; 2: $0,25 < x < 0,5$ ; 3: $0 < x < 0,25$	2
5	Failure Avoidance	Jumlah kesalahan pola yang dihindari pada saat desain dan coding dan dibandingkan dengan jumlah kesalahan pola yang disarankan	$X = A/B$ A: Jumlah kesalahan pola yang dapat dihindari B: Jumlah kesalahan pola yang disarankan	$0 < x < 1$ , makin mendekati 1 makin baik, user dapat lebih sering menghindari kesalahan pola	0: $0 < x < 0,25$ ; 1: $0,25 < x < 0,5$ ; 2: $0,5 < x < 0,75$ ; 3: $0,75 < x < 1$	1
6	Mean Recovery Time	Waktu rata-rata sistem untuk menyelesaikan recovery dari recovery awal sebagian	$X = \text{Sum}(T)/B$ T: Waktu untuk recovery sistem software yang down pada tiap kesempatan B: Jumlah kasus yang diobservasi sistem software telah masuk pada recovery	$0 < X$ , makin kecil makin baik	0: $t > 3$ ; 1: $2 < t < 3$ ; 2: $1 < t < 2$ ; 3: $0 < t < 1$	3
7	Restorability	Kemampuan produk untuk restoring dirinya sendiri setelah kejadian tidak normal atau saat request. Kebutuhan restoration contohnya: database checkpoint, transaction checkpoint,	$X = A/B$ A: Jumlah kasus restore yang sukses dilakukan B: Jumlah kasus restore yang diuji tiap requirement	$0 < x < 1$ , makin besar dan mendekati 1 lebih baik, produk lebih mampu untuk restore pada kasus yang didefinisikan	0: $0 < x < 0,25$ ; 1: $0,25 < x < 0,5$ ; 2: $0,5 < x < 0,75$ ; 3: $0,75 < x < 1$	2

Done

Gambar 4.19 Sub Menu Data Kriteria

Untuk melakukan perubahan pada data yang telah dimasukkan, dengan memilih menu Edit. Terdapat tiga sub menu yaitu: faktor kualitas, sub faktor kualitas dan kriteria seperti pada Gambar 4.20.



Gambar 4.20 Menu dan Sub Menu Edit

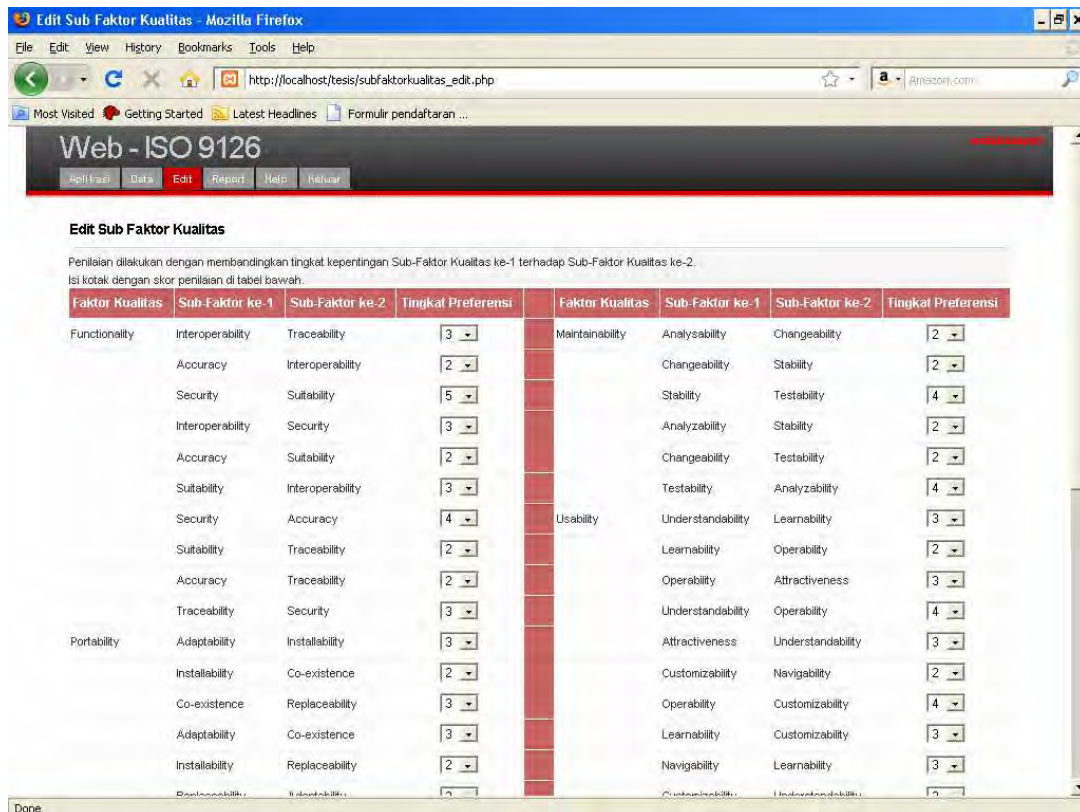
Tiap-tiap form dari sub menu Edit terdapat pada Gambar 4.21 sampai 4.23.

 A screenshot of a web browser window titled 'Edit Faktor Kualitas - Mozilla Firefox'. The address bar shows 'http://localhost/tesis/faktorkualitas\_edit.php'. The page has a header 'Web - ISO 9126' with navigation buttons: 'Aplikasi', 'Data', 'Edit', 'Report', 'Help', and 'Keluar'. The main content area is titled 'Edit Faktor Kualitas' and contains a paragraph: 'Penilaian dilakukan dengan membandingkan tingkat kepentingan Faktor Kualitas ke-1 terhadap Faktor Kualitas ke-2. Isi kotak dengan skor penilaian di tabel bawah.' Below this is a table with three columns: 'Faktor Kualitas ke-1', 'Faktor Kualitas ke-2', and 'Tingkat Preferensi'. The table contains 15 rows of data. At the bottom, there is a text box labeled 'Skor Penilaian'.
 

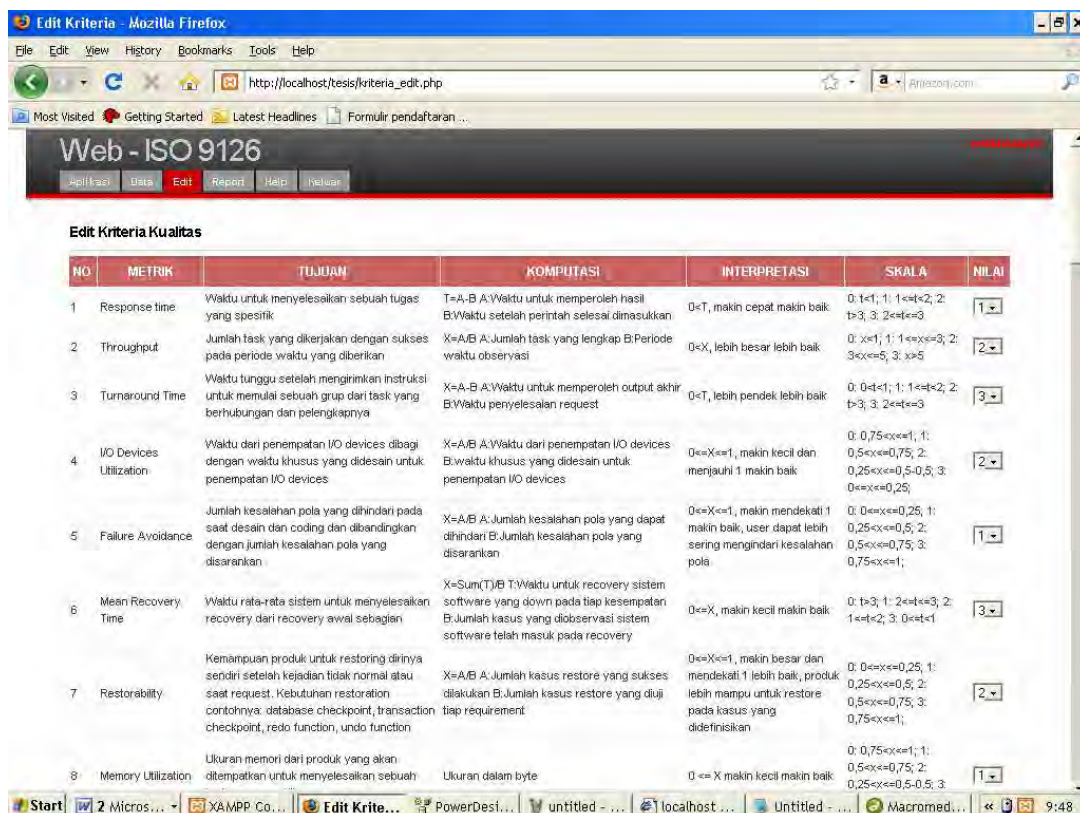
Faktor Kualitas ke-1	Faktor Kualitas ke-2	Tingkat Preferensi
Functionality	Reliability	2
Maintainability	Portability	1
Usability	Efficiency	3
Efficiency	Maintainability	2
Reliability	Portability	5
Functionality	Usability	4
Reliability	Efficiency	3
Efficiency	Portability	1
Maintainability	Reliability	2
Functionality	Efficiency	2
Usability	Maintainability	4
Portability	Usability	4
Maintainability	Functionality	3
Reliability	Usability	5
Functionality	Portability	5

Gambar 4.21 Sub Menu Edit Faktor Kualitas



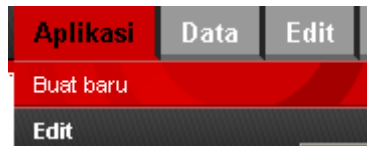


Gambar 4.22 Sub Menu Edit Sub Faktor Kualitas



Gambar 4.23 Sub Menu Edit Kriteria

Untuk melakukan pembuatan aplikasi baru maupun melakukan perubahan terhadap aplikasi yang telah dibuat, dapat dengan menggunakan menu aplikasi seperti pada Gambar 4.24.



Gambar 4.24 Menu dan Sub Menu Aplikasi

#### 4.1.2 Skenario 2

Dilakukan pengisian dengan kondisi terbaik (best case), yaitu semua kriteria diisi nilai sempurna (3). Rekomendasi global diberikan pada Gambar 4.25, sedangkan contoh rekomendasi detail pada Gambar 4.26 untuk faktor kualitas functionality.

Nilai Total Kualitas adalah 100 dengan perincian:

Faktor Kualitas	Nilai	Standar	Grade	Detail
Functionality	51.779	51.779	☆☆☆	<a href="#">Lanjut</a>
Reliability	17.305	17.305	☆☆☆	<a href="#">Lanjut</a>
Usability	5.814	5.814	☆☆☆	<a href="#">Lanjut</a>
Efficiency	5.814	5.814	☆☆☆	<a href="#">Lanjut</a>
Maintainability	17.305	17.305	☆☆☆	<a href="#">Lanjut</a>
Portability	1.984	1.984	☆☆☆	<a href="#">Lanjut</a>

Sub Faktor Kualitas yang harus ditingkatkan kualitasnya:

Gambar 4.25 Rekomendasi Global Skenario 2

Detail Functionality - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address [http://localhost/tesis/detail\\_f.php](http://localhost/tesis/detail_f.php) Go

Web - ISO 9126

Aplikasi Data Edit Report Help Keluar

Nilai Total faktor kualitas Functionality adalah 51.779 dengan perincian:

Sub Faktor Kualitas	Nilai	Standar	Grade
Suitability	8.202	8.20221	☆☆☆
Accuracy	24.354	24.35410	☆☆☆
Security	8.202	8.20221	☆☆☆
Interoperability	8.202	8.20221	☆☆☆
Traceability	2.818	2.81825	☆☆☆

Tindakan yang harus dilakukan untuk meningkatkan kualitas:

[Kembali](#)

Id aplikasi : 00159

Created by Anita H - 2009

Gambar 4.26 Rekomendasi Functionality Skenario 2

### 4.1.3 Skenario 3

Dilakukan pengisian dengan kondisi terburuk (worst case), yaitu semua kriteria diisi nilai 0. Rekomendasi global diberikan pada Gambar 4.27, sedangkan contoh rekomendasi detail pada Gambar 4.28 untuk faktor kualitas functionality.

Nilai Total Kualitas adalah 0 dengan perincian:

Faktor Kualitas	Nilai	Standar	Grade	Detail
Functionality	0.000	51.779		<a href="#">Lanjut</a>
Reliability	0.000	17.305		<a href="#">Lanjut</a>
Usability	0.000	5.814		<a href="#">Lanjut</a>
Efficiency	0.000	5.814		<a href="#">Lanjut</a>
Maintainability	0.000	17.305		<a href="#">Lanjut</a>
Portability	0.000	1.984		<a href="#">Lanjut</a>

Sub Faktor Kualitas yang harus ditingkatkan kualitasnya:

<b>Accuracy:</b>	- Tingkatkan jumlah perhitungan matematika yang dapat diimplementasikan dengan benar - Tingkatkan kebenaran dari spesifikasi fungsi
<b>Suitability:</b>	- Tingkatkan kelengkapan data serta spesifikasi item
<b>Security:</b>	- Tingkatkan kemampuan untuk melakukan pengaksesan audit - Tingkatkan jumlah kebutuhan access controllability yang dapat diimplementasikan secara benar
<b>Interoperability:</b>	- Tingkatkan penggunaan standarisasi untuk protocol dan interfaces - Tingkatkan penggunaan standar pada produk
<b>Fault Tolerance:</b>	- Tingkatkan jumlah kesalahan pola yang dapat dihindari pada saat desain dan coding - Tingkatkan jumlah implementasi fungsi yang memiliki kemampuan menghindari kesalahan operasi

Idaplikasi : 60158

Created by Anita H - 2009

Gambar 4.27 Rekomendasi Global Skenario 3

Detail Functionality - Microsoft Internet Explorer

Address: http://localhost/tesis/detail\_f.php

Web - ISO 9126

Nilai Total faktor kualitas Functionality adalah 0.000 dengan perincian:

Sub Faktor Kualitas	Nilai	Standar	Grade
Suitability	0.000	8.20221	
Accuracy	0.000	24.35410	
Security	0.000	8.20221	
Interoperability	0.000	8.20221	
Traceability	0.000	2.81825	

Tindakan yang harus dilakukan untuk meningkatkan kualitas:

1. Tingkatkan jumlah perhitungan matematika yang dapat diimplementasikan dengan benar
2. Tingkatkan kemampuan untuk melakukan pengaksesan audit
3. Tingkatkan jumlah kebutuhan access controllability yang dapat diimplementasikan secara benar
4. Tingkatkan kemampuan produk untuk diaudit
5. Tingkatkan kejelasan fungsi
6. Tingkatkan penggunaan standarisasi untuk protocol dan interfaces
7. Tingkatkan kelengkapan data serta spesifikasi item
8. Tingkatkan kebenaran dari spesifikasi fungsi
9. Tingkatkan Penggunaan representasi standar data
10. Tingkatkan penggunaan standar pada produk

Kembali

Idaplikasi : 60158

Created by Anita H - 2009

Local intranet

Gambar 4.28 Rekomendasi Functionality Skenario 3

#### 4.1.4 Skenario 4

Dilakukan penilaian suatu aplikasi web dimana yang akan dinilai adalah kriteria complexity. Langkah-langkah yang dilakukan:

1. Mengisi default dari tingkat preferensi faktor dan sub faktor kualitas yaitu 1
2. Mengisi nilai dari semua kriteria dengan 3 kecuali complexity diisi 0

Rekomendasi global diberikan pada Gambar 4.29, sedangkan rekomendasi detail pada Gambar 4.28 untuk faktor kualitas maintainability yang nilainya belum sesuai standar.

**Form Rekomendasi - Microsoft Internet Explorer**

Address: <http://localhost/tesis/rekomendasi.php>

<b>Nama Aplikasi</b>	anandaceria.com
<b>Versi</b>	1.2.3
<b>Kegunaan</b>	Berjualan busana muslim anak
<b>Tanggal Pembuatan</b>	1 Agustus 2009
<b>Developer</b>	cv.andnpg
<b>Keterangan</b>	

Nilai Total Kualitas adalah 98.002 dengan rincian:

Faktor Kualitas	Nilai	Standar	Grade	Detail
Functionality	51.779	51.779	☆☆☆	<a href="#">Lihat</a>
Reliability	17.305	17.305	☆☆☆	<a href="#">Lihat</a>
Usability	5.814	5.814	☆☆☆	<a href="#">Lihat</a>
Efficiency	5.814	5.814	☆☆☆	<a href="#">Lihat</a>
Maintainability	15.306	17.305	☆☆☆	<a href="#">Lihat</a>
Portability	1.984	1.984	☆☆☆	<a href="#">Lihat</a>

Sub Faktor Kualitas yang harus ditingkatkan kualitasnya:

<b>Changeability:</b>	- Tingkatkan kompleksitas dari software
-----------------------	---

Id aplikasi : 60157  
Created by Anita H - 2009

Gambar 4.29 Rekomendasi Global Skenario 4



Sub Faktor Kualitas	Nilai	Standar	Grade
Analysability	6.312	6.31172	☆☆☆
Changeability	3.998	5.99726	☆☆
Stability	2.333	2.33261	☆☆☆
Testability	2.663	2.66341	☆☆☆

Tindakan yang harus dilakukan untuk meningkatkan kualitas:

1. Tingkatkan kompleksitas dari software

[Kembali](#)

Gambar 4.30 Rekomendasi Maintainability Skenario 4

## 4.2 Analisa Uji Coba

Dari hasil uji coba pada skenario 1, terbukti bahwa CASE Tool telah dapat berjalan sesuai dengan tujuan dari penelitian. Rekomendasi diberikan kepada kriteria dari sub faktor kualitas yang memiliki bobot faktor kualitas tertinggi yang sangat berpengaruh pada nilai kualitas total sistem, yaitu functionality dan reliability. Jadi jika ada salah satu yang nilainya belum sempurna (3) akan menjatuhkan nilai kualitas total sistem secara keseluruhan.

CASE Tool juga dapat memberikan penilaian pada kondisi terbaik (skenario 2) maupun kondisi terburuk (skenario 3). Untuk kondisi terbaik, nilai total kualitas mencapai 100. Rekomendasi global maupun detail kosong, karena semua nilai kriteria sudah sempurna (3).

Sedangkan untuk kondisi terburuk, nilai total kualitas adalah 0. Pada rekomendasi global diberikan lima sub faktor kualitas yang harus ditingkatkan kualitasnya beserta tindakan perbaikannya karena memiliki bobot tertinggi. Untuk rekomendasi detail tiap faktor kualitas, berisi semua tindakan untuk meningkatkan nilai kriteria kualitas karena semua kriteria nilainya 0.

Pada skenario 4, CASE Tool memberikan penilaian hanya pada satu kriteria. Terlihat bahwa rekomendasi global dan detail berisi tindakan yang sama yaitu untuk meningkatkan kriteria complexity.

Dari keempat skenario yang telah diujicoba, terlihat bahwa semakin besar bobot dari sub faktor kualitas, maka semakin tinggi pengaruhnya terhadap nilai kualitas total. Sehingga jika salah satu atau bahkan beberapa kriteria yang merupakan anggota dari sub faktor kualitas tersebut belum mencapai nilai sempurna, maka diberikan rekomendasi berupa tindakan yang harus dilakukan untuk meningkatkan nilainya. Rekomendasi diurutkan berdasarkan pada tingkat kepentingan (urgenitas). Rekomendasi dapat diberikan secara detail untuk tiap faktor kualitas karena adanya pemetaan kriteria kualitas aplikasi web ke sub faktor kualitas dari ISO 9126.

## **BAB 5**

### **KESIMPULAN DAN SARAN**

#### **5.1 Kesimpulan**

Dari uji coba yang telah dilakukan, maka dapat ditarik kesimpulan berikut:

1. CASE Tool memberikan panduan bagi pengguna dalam melakukan penilaian terhadap kriteria kualitas berupa kolom tujuan, komputasi, interpretasi, skala dan skala penilaian yang akan mengarahkan pada penilaian yang terukur.
2. CASE Tool melakukan pembobotan faktor dan sub faktor kualitas dengan menggunakan FAHP. Pembobotan bersifat dinamis untuk tiap masukan pengguna terhadap tingkat preferensi faktor dan sub faktor kualitas. Dengan menjumlahkan perkalian bobot sub faktor kualitas dengan nilai kriteria diperoleh nilai total kualitas sistem aplikasi.
3. CASE Tool memberikan dua macam rekomendasi untuk perbaikan kualitas. Rekomendasi global diberikan dengan memperhatikan semua sub faktor kualitas. Sedangkan untuk detail rekomendasi diberikan untuk tiap faktor kualitas. Rekomendasi berupa tindakan-tindakan yang harus dilakukan dengan urutan tingkat kepentingan.
4. Penelitian ini memetakan kriteria kualitas aplikasi web ke sub faktor kualitas dari ISO 9126 sehingga dapat menilai kualitas sistem aplikasi dengan lebih detail sampai ke level sub faktor kualitas.

#### **5.2 Saran**

Untuk meningkatkan hasil yang telah dicapai dari penelitian ini dapat dilakukan beberapa perbaikan sebagai berikut:

1. Data kriteria kualitas pada penelitian ini belum lengkap, maka untuk penelitian selanjutnya dapat lebih dilengkapi.
2. Acuan penilaian tingkat keberhasilan CASE Tool belum ada, sehingga membutuhkan benchmark yang dapat digunakan untuk perbandingan.
3. Dapat ditambahkan kemampuan dari CASE Tool agar mampu mengantisipasi jika nantinya terjadi perubahan pada ISO 9126.

## DAFTAR PUSTAKA

- Behkamal, B., Kahani, M., Akbari, M.K. (2008), "Customizing ISO 9126 Quality Model for Evaluation of B2B Applications", *Information and Software Technology, Elsevier*.
- Boehm, B. W., Brown, J. R., Lipow, M. L. (1976), "Quantitative Evaluation of Software Quality", *Proceedings of the 2nd international conference on Software engineering*, San Fransisco, California, United States, 592-605, IEEE Computer Society Press.
- Botella, P., Burgues, X., Carvallo, J.P., Franch, X., dan Quer, C. (2004), "Using Quality Models for Assessing COTS Selection", *Proceeding of MPEC'04 and ICSE'04*.
- Crosby, Philip B. (1979), *"Quality is Free, The Art of Making Quality Certain"*, New York: Mentor, New American Library.
- Dromey, R.G. (1995), "A Model for Software Product Quality", *IEEE Transactions on Software Engineering* 21.
- Firesmith, D. (2003), "Using Quality Models to Engineer Quality Requirements", *Journal of Object Technology*, Vol. 2, No. 5, September - October, Software Engineering Institute, U.S.A
- Fitzpatrick, R. (1996), "Software Quality Definitions and Strategic Issues", *Technical Paper*, Staffordshire University.
- Grady, R. B. (1992), *Practical Software Metrics For Project Management And Process Improvement*, Prentice Hall.
- Humphrey, Watts, S. (1994), "A Personal Commitment to Software Quality 1", The Software Engineering Institute Carnegie Mellon University Pittsburgh, PA December 1994 issue of *Ed Yourdon's American Programmer Journal*, which focused on "Peopleware."
- IEEE Std 610.12-1990, *Glossary of Software Engineering Terminology*, Institute of Electrical and Electronics Engineers, Inc., 1990
- ISO/IEC 9126-1, *Information technology - Software Quality Characteristics and Metrics - Part 1: Quality Characteristics and Subcharacteristics*, International Organization for Standardization, International Electrotechnical Commission, Geneva, 1991.
- ISO/IEC 9126-1, *Software Engineering – Product Quality – Part 1: Quality Model*, first ed.: 2001-06-15.
- Khosravi, K., Gueheneuc, Y. (2004), "A Quality Model for Design Patterns", *Technical Report 1249*, University of Montreal.
- Khosravi K., Gu'eh'eneuc (2005), Y. "On Issues with Software Quality Models", 19th European Conference on Object-Oriented Programming SECC
- McCall, J. A., Richards, P. K., & Walters, G. F. (1977). *Factors in Software Quality*. Griffiths Air Force Base, N.Y. : Rome Air Development Center Air Force Systems Command.
- Petrash, Roland (1999), *"The Definition of Software Quality: A Practical Approach"*. petrasch@computer.org
- Pressman, R. (2000), *Software Engineering, a Practitioner's Approach*, fifth edition, McGraw-Hill.
- Schulmeyer, Gordon, G., James I. McManus (1998), *"Handbook of Software Quality Assurance 3<sup>rd</sup> ed."*, Prentice Hall.



- Setiyoko, A.S., et al. (2005), "Pendekatan Fuzzy AHP dan Fuzzy MCDM Untuk Pengalokasian Fasilitas", *Prosiding Seminar Nasional Manajemen Teknologi I*, 25-26 Pebruari 2005, Institut Teknologi Sepuluh Nopember, Surabaya
- Turban, E., et al. (2004), *Electronic Commerce: a Managerial Perspective*, Prentice Hall.
- Wahono, R.S., "Teknik Pengukuran Kualitas Perangkat Lunak", dalam RomiSatriaWahono.Net, 2006
- Zwass, V., "Electronic Commerce: Structures and Issues", *International Journal of Software Quality* 1, 1996.